2

LABORATORY FOR
COMPUTER SCIENCE

MASSACHUSETTS
INSTITUTE OF
TECHNOLOGY

MIT/LCS/TR-549

WORD AND SUBWORD MODELLING IN A SEGMENT-BASED HMM WORD SPOTTER USING A DATA ANALYTIC APPROACH

[redacted]

Jeffrey Neil Marcus

September 1992

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE

Word and Subword Modelling in a Segment-Based HMM
Word Spotter Using a Data Analytic Approach

5. FUNDING NUMBERS

6. AUTHOR(S)

Marcus, J. N.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MIT, Laboratory for Computer Science
545 Technology Sqaure
Cambridge, MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

MIT/LCS/TR-549

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DARPA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In this work we focus on methods for representing acoustic-phonetic knowledge in a speech recognizer and for analyzing the system's behavior in detail. The testbed for developing these methods is a segment-based hidden Markov model (HMM) recognizer. In this system, measurements are made on variable-duration segments. Ideally, each segment is associated with a single phonetic unit, which we refer to as a phone. The scheme has several potential advantages over the typical HMM recognizer, which is based on fixed-duration frames. They include a greater ability to model statistical dependence among spectral measurements, a more convenient framework for representing acoustic-phonetic knowledge, and a potential reduction in computation since the mean segment rate in our implementation is 1/5 of a typical frame rate.

The HMM framework is used to model the segmenter's deviations from the ideal behavior of one segment per phone. We employ an HMM topology that allows a phone to be associated with more than one segment. Biphone HMM's model instances in which a segment is associated with more than one phone.

We compared the effectiveness of various segment measurement sets on a phonetic recognition task. The measurements consisted of short-time spectral representations measured at particular positions relative to segment boundaries. The key result was that the addition of spectra measured outside the segment to those measured inside led to a significant improvement in performance. For the task of recognizing 39 phone labels, the best system attained a phonetic accuracy (% correct - % insertions) of 59% (95% confidence interval of 53-65%) on a set of nine male speakers from the VOYAGER corpus, a result in the range of those previously reported for recognizers of comparable complexity.

15. NUMBER OF PAGES

319

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

Word and Subword Modelling in a Segment-Based HMM Word Spotter Using a Data Analytic Approach

by

Jeffrey Neil Marcus

S.M., Massachusetts Institute of Technology
(1984)

S.B., Massachusetts Institute of Technology
(1982)

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

September, 1992

Signature of Author ....
Department of Electrical Engineering and Computer Science
July 28, 1992

Certified by ....
Victor W. Zue
Thesis Supervisor

Accepted by ....
Campbell L. Searle
Chair, Department Committee on Graduate Students

Word and Subword Modelling in a Segment-Based HMM Word Spotter Using a Data Analytic Approach

by

Jeffrey Neil Marcus

Submitted to the Department of Electrical Engineering and Computer Science on July 27, 1992 in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

In this work we focus on methods for representing acoustic-phonetic knowledge in a speech recognizer and for analyzing the system's behavior in detail. The testbed for developing these methods is a segment-based hidden Markov model (HMM) recognizer. In this system, measurements are made on variable-duration segments. Ideally, each segment is associated with a single phonetic unit, which we refer to as a phone. The scheme has several potential advantages over the typical HMM recognizer, which is based on fixed-duration frames. They include a greater ability to model statistical dependence among spectral measurements, a more convenient framework for representing acoustic-phonetic knowledge, and a potential reduction in computation since the mean segment rate in our implementation is 1/5 of a typical frame rate.

The HMM framework is used to model the segmenter's deviations from the ideal behavior of one segment per phone. We employ an HMM topology that allows a phone to be associated with more than one segment. Biphone HMM's model instances in which a segment is associated with more than one phone.

We compared the effectiveness of various segment measurement sets on a phonetic recognition task. The measurements consisted of short-time spectral representations measured at particular positions relative to segment boundaries. The key result was that the addition of spectra measured outside the segment to those measured inside led to a significant improvement in performance. For the task of recognizing 39 phone labels, the best system attained a phonetic accuracy (% correct - % insertions) of 59% (95% confidence interval of 53-65%) on a set of nine male speakers from the VOYAGER corpus, a result in the range of those previously reported for recognizers of comparable complexity.

In the course of investigating methods for representing knowledge in the measurement sets, we built linear regression models to estimate F_1 and F_2 from a set of mel-frequency spectral coefficients (MFSC's). We show that such a model is inadequate for predicting formant values at the ends of their observed ranges. However, by adding nonlinear transformations of the MFSC's to the regressor set, highly-accurate models ($R^2 > .96$) valid for more than 80% of observed formant frequencies could be built.

We employed multiple discriminant analysis to reduce the dimensionality of the measurement sets. We also developed a technique we term *grouped multiple discriminant analysis* to address the fact that within-class covariance varies greatly among phones, contrary to the assumptions of conventional multiple discriminant analysis. By clustering within-phone covariance matrices hierarchically using a distance metric based on a statistical test for the equality of covariances, we show that they cluster well by phone type. Grouped multiple discriminant analysis attempts to exploit this fact. Phonetic recognition performance using this method to reduce dimensionality was near that of conventional multiple discriminant analysis.

We used the segment-based HMM to investigate word modelling issues as well. Models were compared using a word spotting task. The models varied along three dimensions: training method, type of pronunciation network, and measurement set. Even for fairly small training sets, models trained from word-specific data outperformed those built from context-independent subword models. Multiple-pronunciation networks were shown to be superior to single-pronunciation networks for modelling pronunciation and segmenter variability. Finally, as was the case for phonetic recognition, word spotting performance improved when out-of-segment measurements were added to the set.

In the course of this investigation, we developed novel algorithms for word spotter scoring and performance evaluation. The scoring algorithm determines the beginning and end points of a presumed keyword and computes an estimate of the probability that the keyword occurred between those points. The performance evaluation algorithm is novel in that it breaks speech into discrete "trials" and counts the number of trials corresponding to false alarms and correct detections as a function of a keyword detection threshold. This turns out to be a convenient framework for both overall performance evaluation and detailed error analysis.

Finally, we outline the philosophy of exploratory data analysis and discuss how the methodology can be employed in the design of speech recognizers. To demonstrate the approach, we develop a set of interactive graphical tools for diagnosing recognition errors at various levels of detail. At the most detailed level, the tools assign "blame" for an error to individual measurements. We demonstrate the tools in case studies of word spotting errors. By using these tools to gain insight into recognizer behavior, a system designer will presumably be able to diagnose and remedy acoustic modelling deficiencies. This will most likely be possible if a large number of errors can be attributed to a few deficiencies. For the case studies, we show that most errors do indeed seem to be related to a small number of underlying deficiencies..

Thesis Supervisor: Dr. Victor W. Zue
Title: Principal Research Scientist

Acknowledgments

I wish to thank Victor Zue, my thesis supervisor, for the guidance he has provided me over the past five years. He has been a great teacher, mentor and friend. As well as teaching me a great deal about speech and about doing research, he has always taken an active interest in my academic and professional advancement. Also, he has built a group of talented and congenial people, who have made working here an enriching and enjoyable experience. Finally, in all his dealings with me, he has shown great compassion and generosity. I hope that I can follow his example if I am ever in the position to guide students in the years to come.

I would like to thank the members of my thesis committee, Lou Braida and Ken Stevens, for their suggestions and interest in my work. I have greatly enjoyed working with both of them and the thesis has benefited from their involvement. In particular, the algorithm used to evaluate word spotting performance was inspired from a discussion with Lou. Ken's suggestions concerning where to make acoustic measurements led to much of the work of Chapter 4. More generally, discussing speech with Ken has always been thought-provoking and an important part of my education.

I have also benefited from the opportunity to work with a talented group of researchers. By looking over Jim Glass's and Mike Phillips' shoulders, using their software and talking to them, I have learned much about segment-based approaches and intensive data analysis, two major themes of the work. Mark Randolph introduced me to the concept of applying exploratory data analysis to speech. Discussions with Hong Leung and Lee Hetherington have helped me clarify aspects of the work. In particular, Lee made valuable suggestions concerning the word spotting evaluation algorithm.

Thanks to Caroline Huang for beginning my formal education in spectrogram reading and the members of the spectrogram reading group, Caroline, Nancy Daly, John Pitrelli, Jim Glass, and Hong Leung for continuing that education.

Many past and present members of the Speech Communications and Spoken Language Systems groups have helped me get things done. In particular, Mike Phillips has provided me with much assistance and software over the years as I struggled to learn about the Lisp machines and UNIX. I am also indebted to the following people for their software: Jim Glass for the multi-level segmentation, Kevin Yu for the mel-frequency spectral coefficient computation, Hong Leung for vector quantization and Rob Kassel for transcription alignment. Dave Kaufman, Katy Isaacs, Joe Polifroni and David Goodine have kept the Lisp machines and Suns going. Rob Kassel's hard work has allowed document preparation on the Macintosh to be relatively painless. Mike Phillips and Lee

Hetherington have done much to improve the UNIX software environment. Finally, over the years, Arlene Wint, Christie Winterton, and, especially, Vicky Palay, have administered this group in an incredibly efficient and friendly manner. Thank you.

It has been my good fortune to share an office over the past two and a half years with Lee Hetherington. On top of being a good sounding board for my ideas, he has been my UNIX and C guru. He has almost always had an immediate answer to my questions and when he has not had an immediate answer, he has often worked until he has found one. Moreover, he has listened in good humor for many hours to my incessant praise and occasional complaints about S-Plus. It has been a pleasure to have him as an officemate.

Thanks to Caroline Huang and Rob Kassel for taking part of a Sunday to listen to a dry run of my thesis defense presentation.

I am proud to call many of the past and present members of both the Spoken Language System and Speech Groups my friends. They have made the rough times easier and have been a lot of fun to work with. They know who they are.

Thanks to Matt Lennig and Paul Mermelstein, my managers at Bell-Northern Research, for exciting my interest in speech and for making it possible for me to receive a very generous scholarship from Bell-Northern.

Thanks to my family, for all their love and support.

My deepest feelings of gratitude and love are reserved for my wife, Susan. She has made many sacrifices over the years, uprooting herself to join me on this quest, going through an arduous medical internship for the second time, and living with an often-absent and sometimes bad-tempered husband. She constantly encouraged me when I was down, especially as the time to complete the thesis kept expanding. Having her with me has been my greatest source of joy the past five years. This thesis is dedicated to her.

Finally, I thank Caleb, whose birth three months ago put it all in perspective.

This research was supported by Bell-Northern Research, FCAR (Quebec), NSERC (Canada) and DARPA-SISTO.

To Susan

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The prevalent approach to automatic speech recognition involves the use of hidden Markov models (HMM's) to model probability distribution functions (PDF's) of measurements made on fixed-length segments, or frames, of speech.[1] In their most basic form, the measurements made on each of these frames represent the magnitude of the short-term spectrum in the vicinity of the frame. A good description of an early system of this sort appears in [Rabiner 83]. An excellent review of advances in speech recognition up to 1988 appears in [Lee 88]. For more detailed descriptions of speech recognition systems built before 1980 and reviews of the history of speech recognition to that point, see [Lea 80].

More recently, state-of-the-art continuous speech recognizers, e.g., [Chow 87, Cohen 90, Paul 91], have taken advantage of increased computing power and access to more training data to build HMM recognizers that address certain modelling deficiencies of earlier versions. These can be classified as either acoustic-phonetic or statistical modelling deficiencies. Examples of the acoustic-phonetic modelling deficiencies include the failure to account for the effect of coarticulation on the realization of a given phoneme, which has been addressed with triphone and/or word-specific phone modelling [Bahl 80, Schwartz 85, Chow 86, Lee 88]; and the inability to deal with speaker variability, which has been addressed with gender-specific models [Kubala 91], vari-

[1] A guide to abbreviations used in the thesis appears in Appendix C

ous speaker-adaptation schemes [Stern 87, Rigoll 89, Rtischev 89, Kubala 90, Stern 91, Bellegarda 92] and the introduction of measurements of spectral change [Furui 86] that vary less among different speakers than do static measurements. Examples of statistical modelling deficiencies include weaknesses in vector quantization [Rabiner 83] which have been addressed by more effective techniques that use multiple VQ codebooks [Gupta 87, Chow 87, Lee 88, Cohen 90] or mixture distributions [Huang 89, Bellegarda 90, Ney 90, Paul 91]; and incorrect assumptions about the form of the PDF, which has been addressed with discriminant or connectionist training techniques, e.g., [Brown 87, Katagiri 91, Renals 92] and postprocessors of various types (summarized in [Schwartz 92]). Finally, other researchers have eschewed the HMM framework altogether and have adopted other approaches. These include the stochastic segment [Ostendorf 89, Zue 89a] and segmental neural network [Leung 92] approaches, which have also attempted to deal with various perceived deficiencies in HMM's. We might also include in this inventory approaches such as neural networks [Waibel 88, Leung 89, Lippmann 89] which have been used mainly for speech recognition subtasks such as phoneme recognition rather then as the bases for complete systems.

Published reports of system improvements tend to indicate that the model deficiencies addressed were anticipated based on prior knowledge about acoustic-phonetics and statistical modelling rather then analysis of the behavior of the system being improved. Also, while speech recognizer design has been informed by general knowledge about perception (e.g., the commonly used mel-frequency representation [Davis 80] is based on the psychophysical concept of critical-band filters [Zwicker 61]) and acoustic-phonetics (e.g., triphone modelling is a response to the effect of coarticulation), little attempt has been made to examine how this knowledge, as well as more detailed knowledge about speech, is represented in the Markov models so as to develop insight into model deficiencies. Instead, researchers have generally improved a speech recognizer by recognizing an existing deficiency based on prior knowledge, attempting to remedy the deficiency, usually by increasing system complexity, and testing whether the improvement had the desired effect with general measures of recognizer performance, such as overall word error rate.

While this methodology has been successful in reducing error rates dramatically (e.g., Lee [Lee 90] cites word error reductions of greater than 50% in going from phone to triphone models), it is not at all clear that the methodology, along with increased training data and increased number of system parameters for taking advantage of the extra data, will continue to lead to increased performance indefinitely, as has been suggested by some [Lee 89a]. We believe that at some point, a deeper understanding of the system will be required to effect improvement. Put another way, we feel it is unjustified to assume a recognizer's strengths and weaknesses without analyzing recognizer errors and tracing their sources. Thus, we believe a new methodology, based on understanding recognizer behavior in more detail and relating the behavior to our prior knowledge about acoustic-phonetics and statistics, will be required for diagnosing and remedying model deficiencies.

This belief motivates our work, whose primary goals are to develop a better understanding of how acoustic-phonetic knowledge is represented in a statistical speech recognizer and to develop a methodology for analyzing system behavior at a deep level in terms of this knowledge. Towards these ends, we make heavy use of the philosophy of exploratory data analysis (EDA) propounded by Tukey [Tukey 77] and others.

We develop this understanding in the context of building an HMM recognizer. We use the HMM approach because it is a powerful statistical modelling technique with fairly simple mathematical underpinnings compared to those of competing statistical approaches such as neural networks [Lippmann 89] and stochastic segment models [Ostendorf 89, Zue 89a]. For these reasons, it is well-suited to our goal of attaining deeper understanding of system behavior.

However, our system is atypical of most HMM implementations in that it makes measurements on variable-duration segments produced by a segmentation algorithm rather then on the fixed-length frames commonly used. The segments are delineated by acoustic landmarks across which there are large spectral changes, and these landmarks often correspond to acoustic-phonetic boundaries. Such a system has certain attractive features compared to a frame-based one including a better framework for modelling correlations among neighboring spectra, less computation, and, perhaps most important for our work, a more appropriate framework for incorporating acoustic-phonetic knowledge. We discuss these features in more detail in Chapter 3, where the segment-based HMM system is described. We know of only one implementation of this type [LeMaire 89] and it was tested on a digit recognition task. Thus, there has been little discussion of how to build subword models in such a system and no published evaluation of such a system on a more difficult task. A secondary goal of our work is to illuminate the issues and to evaluate the feasibility of this approach to speech recognition.

Another goal is to investigate several issues that arise in building acoustic models of words. In particular, we compare training strategies and pronunciation network topologies. We conduct this investigation in a word spotting framework and suggest certain refinements to the methods currently used for scoring and evaluation of word spotters.

1.1 Previous Work on Speech Knowledge Representation and Error Analysis in Statistical Speech Recognizers

As pointed out above, all existing statistical speech recognition systems incorporate some general knowledge about speech in an implicit manner. However, while the extent to which more explicit and detailed representation of speech knowledge in a statistical recognizer can improve recognition has been dealt with by several researchers, it remains an open question.

Zue [Zue 85] advocated increased use of speech knowledge to improve existing algorithms. Two areas of application were identified: the selection of acoustic attributes for characterizing the signal and the control strategy used for combining acoustic attributes so as to recover the underlying utterance. Zue pointed to some successes in applying speech-specific knowledge to the selection of acoustic attributes but pointed out that there had been little success in applying speech knowledge to the cue integration problem and that classical multivariate statistical techniques worked best in practice. He attributed the success of these and other statistical techniques to their ability to model ignorance about the underlying process and to thereby make "optimal use of what knowledge we do have" rather then to any theoretical justification for their use.

Conversely, Levinson [Levinson 85] argued that statistical methods "constitute a powerful theory of speech that can be reconciled with and elucidate conventional linguistic theories while being used to build truly competent mechanical speech recognizers" and provided a comprehensive review of these methods. Just as Zue pointed out that there was some merit in using statistical methods, Levinson pointed out that there was some merit in using speech-specific knowledge. Specifically, he argued that "regularities which have been catalogued by linguists are too striking and consistent to be accidental and should be used to constrain the [statistical] models" which include variables that "merely account for non-information-bearing artifacts." Reducing the number of degrees of freedom in these models, would presumably improve model parameter estimation for a fixed amount of training data.

With some exceptions, the evolution of speech recognition research since these views were published in 1985 has been away from the use of explicit speech knowledge. In fact, instead of using such knowledge even minimally so as to constrain model complexity, as proposed by Levinson, models have become ever more complex and have relied on increased training data rather then explicit speech knowledge to improve parameter estimation of the more complex models.

Counterexamples to this trend, consisting of attempts to incorporate speech knowledge explicitly, have had mixed results. We present several such examples here. We will confine our examples to continuous or connected speech recognition systems since they provide more stringent tests of the effectiveness of explicit knowledge representation than do simpler tasks.

Encouraging results were obtained by Bush and Kopec [Bush 87] who achieved among the best results of that time for speaker-independent connected digit recognition using a recognizer that made extensive use of explicit knowledge in selecting acoustic measurements and in accounting for phenomena such as prepausal lengthening. Moreover, in comparing different system configurations, they found that configurations that made use of this knowledge performed best. The measurements were made on variable-length acoustic segments rather then fixed-length frames and so were not limited to short-term spectral representations and measurements of short-term spectral change. Thus, this framework was more amenable to the inclusion of knowledge-based measurements.

However, it should be pointed out that since the publication of that paper, better results have been obtained on the same task using frame-based HMM's that include little speech-specific knowledge [Doddington 89]. Thus, for this task at least, it cannot be said that speech-specific knowledge has been useful.

Lee [Lee 88] found that certain additions of speech-specific knowledge to SPHINX, a frame-based HMM for speaker-independent continuous speech, improved performance while others did not. In particular, system performance improved when mel-frequency cepstral coefficients [Davis 80] were used instead of LPC-based cepstral coefficients and when differenced cepstral coefficients were added to the measurement set. Also, the use of phonological rules to transform pronunciation baseforms led to some improvement. However, an attempt to integrate "knowledge-based" measurements based on variable-length acoustic segments rather then frames into the HMM by combining acoustic scores from a system based on such measurements with those of SPHINX degraded performance. Also, performance was about the same for a system that used explicit pronunciation networks and multiple pronunciations for capturing phonological variability at the word level as for one that used a single pronunciation per word. Thus, as Lee pointed out, "the greatest improvements came not from intricate acoustic-phonetic parameters or elaborate phonological rules, but from crude knowledge of speech and English."

Conversely, papers by SRI International [Weintraub 89, Cohen 90] report that the incorporation of explicit phonological rules improved performance in a high-performance HMM speech recognizer. The authors attributed the difference between their results and Lee's to the fact that the networks used in their work allowed relatively few pronunciations per word compared to those of the multiple-pronunciation networks used by Lee, and thus provided a more appropriate degree of constraint on allowable pronunciations [Weintraub 89]. The SRI results suggest that speech-specific knowledge can be useful, at least at the phonological level.

There have been a few other attempts to apply explicit knowledge to the selection of acoustic measurements in a statistical speech recognizer. One of these has been the SUMMIT speaker-independent continuous speech recognition system [Zue 89a], which uses a semiautomatic method [Zue 89b] to determine measurements made on variable-length acoustic segments. The measurements are based on explicit speech-specific knowledge rather then being based exclusively on short-term spectral representations. For instance, certain of them are intended to model formant frequencies, since formants play a large role in theories of speech production and perception. The most recent comparisons of the performance of SUMMIT to frame-based HMM recognizers on the same task indicate that performance of SUMMIT is comparable but slightly inferior to the best HMM recognizers [Pallett 91]. Since SUMMIT's design is not as mature as that of frame-based HMM systems, ongoing improvements in SUMMIT may change this situation.

Still, it has yet to be demonstrated that a recognizer that uses explicit speech-specific knowledge in determining acoustic measurer·ents can outperform one that uses solely short-term spectra within an HMM framework. One possible explanation for this is that frame-based HMM's implicitly characterize many of the same acoustic phenomena that advocates of speech-specific knowledge believe should be characterized using a more complex set of attributes and/or a more complex control strategy. It would be worthwhile determining the extent to which this is true. Armed with this knowledge, a speech recognizer designer could focus future attempts at incorporating explicit knowledge on the modelling of phenomena that are not well characterized by frame-based HMM's. As discussed in the next section, portions of our work are concerned with this investigation. We investigate the issue directly by building models relating short-term spectra to more speech-specific attributes such as formants and distinctive features in Chapter 4 and indirectly by developing methods of analyzing recognizer errors at a detailed level in Chapter 6.

In contrast to the issue of speech-specific knowledge, detailed error analysis has received little attention by speech recognition researchers. Usually, oublished reports of system performance include only general measures such as word and sentence error rate with little attempt made to analyze error patterns in greater detail. However, there have been some exceptions. In some cases, e.g., [Lee 88], errors have been enumerated and some attempt has been made to find patterns in them. Doddington [Doddington 89] made a relatively detailed error analysis on a connected digit recognition task, including an analysis of the types of confusions made, an identification of the characteristics of speakers for which there were high error rates, and spectrographic displays and LPC syntheses of several of the system's acoustic models. In neither of these cases was the published error analy 's used as a basis for making further system refinements. However, Bush and Kopec [Bush 87] examined confusion matrices and spectrograms of misrecognized tokens and used the results of these examinations to refine their system, thus demonstrating the value of detailed error analysis.

1.2 Scope and Structure of Thesis

We use the VOYAGER [Soclof 90] and TIMIT [Lamel 86] speech corpora in our investigations. Both include continuous speech from multiple speakers. The TIMIT corpus is used for training only while the VOYAGER corpus is used for both training and testing. In Chapter 2 we describe these corpora and how they are used in our work in more detail. Also, we describe the two frequency schemes we use to characterize short-term spectra: hair-cell envelopes and mel-frequency spectral coefficients. The acoustic measurements we make throughout our work are based on these spectral representations. Finally, we provide a brief description of principal component analysis, a technique used to reduce the dimensionality of the spectral representations.

In Chapter 3, we describe the segmentation algorithm and motivate its use in more detail. We then provide an analysis of segmenter behavior. Finally, we discuss how subword HMM's are built out of the segments, paying special attention to design issues that are specific to the segment-based HMM approach.

The issue we discuss in greatest detail is that of designing a set of acoustic measurements for characterizing each segment. This matter is dealt with in Chapter 4. The measurement sets considered are compared on a phonetic recognition task that uses the subword models introduced in Chapter 3. Our motivation for investigating different measurement sets is twofold. First of all, it is primarily through the measurements that acoustic-phonetic knowledge can be incorporated in the recognizer. Secondly, choosing measurements in a segment-based system is not as straightforward as in a frame-based one because it is not appropriate to assume that the spectrum is stationary over the course of a segment.

We use the phonetic recognition task for the comparison because it is fairly simple to implement and because it has been used by others [Schwartz 85, Lee 89b, Leung 90, Robinson 91b, Digilakis 92] to test various systems and so provided a convenient way to test the feasibility of the segment-based HMM approach.

Chapter 4 contains several results of particular interest:

1. The relationship between mel-frequency spectral coefficients and formant frequencies is shown to be non-linear. We develop a multiple regression model for F_1 and F_2 based on non-linear transformations of the coefficients that is highly accurate, though over a restricted range of formant frequencies.

2. The inclusion in the measurement set of spectral measurements made just outside the segment is shown to improve performance substantially.

3. The best results achieved are comparable to those reported for existing systems of similar complexity. Thus, the segment-based HMM appears to be a feasible alternative for speech recognition to existing approaches.

While the phonetic recognition task is a convenient one for studying certain issues in speech recognition, the ultimate goal of most speech recognition systems is to recognize strings of words. There are certain issues in word modelling that are of interest but cannot be studied in the phonetic recognition task. Thus, in Chapter 5, we incorporate the segmenter in a word spotter, which is a system that spots designated *keywords* in a stream of continuous speech. The word spotting task allows us to concentrate on modelling a few keywords of interest. This contrasts with the word recognition task, for which we would have to model each word in the vocabulary of the speech corpus being used to test the system. This simplification contributes to the goal of understanding system behavior at a detailed level since it pares away complexity that makes such understanding difficult. Within the chapter, we provide a formal description of the word spotting task, introducing several refinements to the techniques currently used for keyword scoring and for performance evaluation, including a significance test for comparing the performance of two word spotters on the same task. Also, we compare the performance of word- and subword-trained models and investigate the effect on performance of the use of word pronunciation networks of various types. Finally, we introduce techniques for analyzing speech recognizer behavior at a detailed level and use them to explain a large increase in performance observed in going from a single- to a multiple-pronunciation model.

In Chapter 6, we outline some of the key principles of exploratory data analysis and introduce a specific methodology for building speech recognizers based on these principles, contrasting it to the prevailing methodology. Additionally, we develop specific techniques for applying these principles to the design of speech recognition systems. In particular, we extend the techniques introduced in Chapter 5 for analyzing recognizer behavior so as to gain insight into behavior at a deeper level. The new techniques are demonstrated in a case study of word spotting errors. The aim of the techniques is to allow the system designer to rapidly identify errors and to develop an understanding of their causes at the measurement set level. It is hoped in using such techniques that the values of a small number of measurements can be blamed for a large percentage of errors and that, moreover, patterns can be identified that will allow system behavior to be interpreted in terms of acoustic-phonetic knowledge. If these hopes are fulfilled, diagnosing model deficiencies becomes tractable since the diagnosis can be conducted in a relatively small measurement subspace, and acoustic-phonetic knowledge can be brought to bear on the problem. For the case study, the hope is fulfilled. While diagnosing the specific model deficiencies that caused the errors is beyond the scope of our work, we provide specific suggestions for pursuing the diagnosis and general suggestions for using the information thus gained to improve the models.

Finally, in Chapter 7, we summarize our work and suggest promising directions for future work.

The remainder of this chapter consists of an outline of a general model for HMM speech recognition. The purposes of the model are to

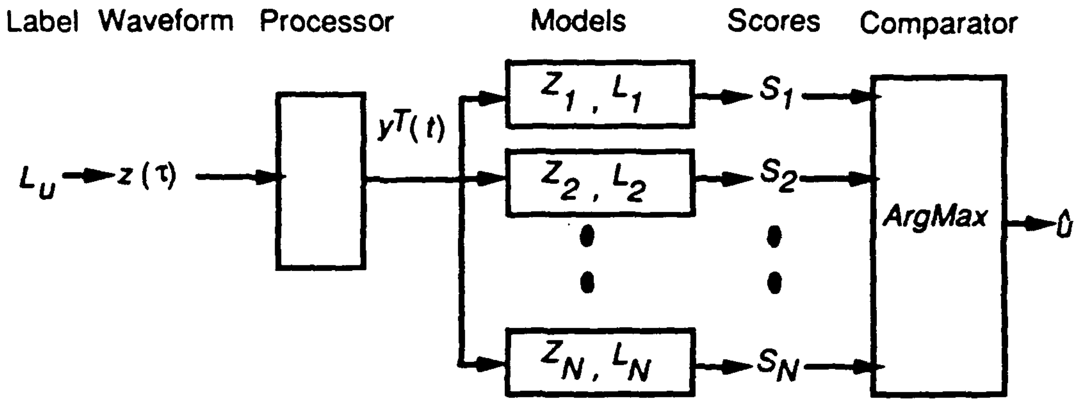


Figure 1.1: A statistical speech recognizer. See text for explanation of symbols.

1. introduce the characteristics common to most HMM speech recognizers,
2. make explicit the assumptions that are usually made implicitly in their design, and
3. provide a framework for describing our work and comparing it to other work in the field.

1.3 The Speech Recognition Model

The following is a general description of an HMM-based speech recognizer. The description presented pertains not only to conventional frame-based HMM's but to the variable-duration segment HMM's used in our work. Many of the aspects presented are common to other statistical speech recognizers so we will tend to be as general as possible in our description, pointing out aspects specific to HMM's when necessary.

The general model is schematized in Figure 1.1. A speaker produces a *test utterance*, which consists of one or more words from a finite vocabulary. The uiterance can be represented as an *utterance label* L_u and an *observed* *waveform* $z(\tau)$ where u is an index into the set of all possible utterances. For the purposes of simplicity, we will assume that there is a finite number of allowable utterances, although this is not true if there are no limits on the number of words in the utterance. The utterance label is typically the orthographic transcription of the string of words produced. The waveform is usually digitized and can thus be represented formally as a sequence of integers. Generally, the waveform is processed and converted into a sequence of *observation vectors* $\boldsymbol{y}^T(t)$, $1 \leq t \leq T$ where T is the total number of vectors in the utterance.[2] We describe this process in greater detail below. The speech recognizer must determine the correct utterance label given the utterance waveform. A statistical recognizer accomplishes this by building *utterance models*. Each such model consists of an hypothesized utterance label L_i and an *utterance acoustic model* Z_i. The latter is a statistical model for the observation vectors given the hypothesized label. The recognizer computes a score S_i that reflects the likelihood that model Z_i produced the sequence of $\boldsymbol{y}^T(t)$. The recognizer outputs the hypothesized index $\hat{u}$ corresponding to the utterance acoustic model $Z_{\hat{u}}$ with the highest score. The recognizer is correct if $\hat{u} = u$.

Most current statistical speech recognizers have other details in common. They each include a set of *lexical-acoustic units*, each unit consisting of a *lexical label* a_i and a *lexical-acoustic model* M_i where i is an index into the set of units. The set of units is chosen by the system designer. For instance, in a small-vocabulary recognizer, each unit is typically a word.

A key characteristic of these models is that they can be concatenated to form utterance acoustic models. In HMM recognizers the method for concatenating two acoustic models is to join the final state of one model to the initial state of the next one [Jelinek 76, Lee 88].

The utterance model is built out of the lexical-acoustic units according to

[2] We will conventionally represent observation vectors as row vectors. This will be convenient for algebraic manipulations of these vectors introduced in later chapters.

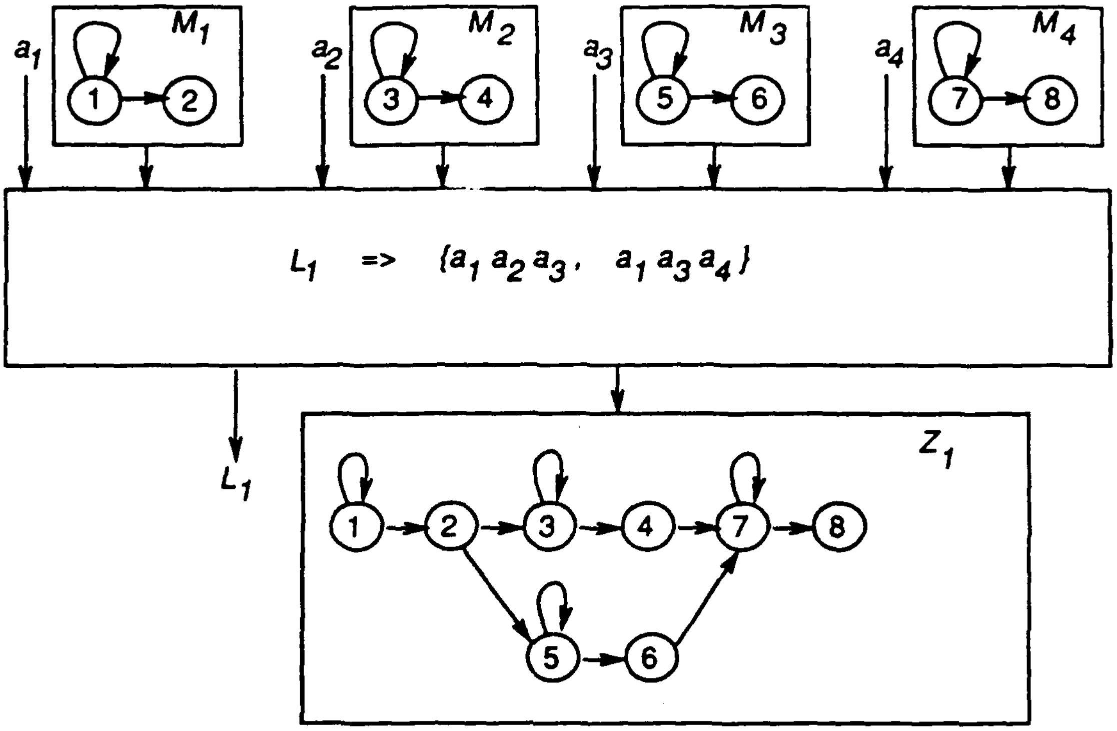


Figure 1.2: Example of lexicon rule. Utterance model Z_1 with label L_1 built out of HMM's M_1, M_2, M_3, and M_4 with corresponding lexical labels a_1, a_2, a_3, and a_4. Circles are HMM states identified by index and arrows are allowable state transitions. Transition probabilities not included. Note that the utterance label is represented by more than one sequence of lexical labels.

rules specified in the *lexicon*. Figure 1.2 represents the building of an utterance model out of constituent lexical-acoustic units.

An example of a simple lexicon is that for a small-vocabulary, isolated word recognizer, for which the lexical-acoustic units model words. For this example, the lexicon specifies a one-to-one mapping between lexical-acoustic models and utterance acoustic models as well as between lexical labels and hypothesized utterance labels.

In large-vocabulary systems, the labelling process is not quite as stra.ghtforward. It is currently infeasible to build a model directly for each word in a large vocabulary because this would require enough instances of each word to generate statistically reliable models. This, in turn, would require the collection of a very large training corpus. While the collection of such large corpora is underway [Phillips 92], at the current time devoting specific models to each word in a large vocabulary system is impractical.

Thus, a smaller set of labels capable of representing each word in the vocabulary must be used. Theoretically, phonemes could be used. However, it is typical to take the surface realization of the underlying phoneme string into account when representing a word in terms of constituent labels so that there is less variation in the acoustic realizations for a given label. Thus, for example, in a continuous-speech system, any vowel that tends to be reduced might be represented by the same label whether or not the underlying phoneme is a /ə/. As another example, distinct labels may be used to represent stops that tend to be released and those that do not. These labelling choices are made by the system designer. We will refer to labels which are based on the surface realization of the underlying phoneme string as *phones* and to their corresponding acoustic models as *phone models*.

In large-vocabulary systems, phone labels are concatenated to form word labels according to rules specified by the lexicon. Utterance labels are built in turn by concatenating word labels. A similar process is used to generate utterance acoustic models from phone models.

The lexicon may include a *pronunciation network* and a *language model* [Jelinek 76]. The former assigns a probability to each mapping from a word label to a lexical label sequence. These probabilities are used to build word acoustic models from the lexical-acoustic models. In an HMM system, this is done by assigning probabilities to arcs joining the final state of one lexical HMM to the initial state of the following one. The language model is analogous to the pronunciation network but is used for building utterance acoustic models out of word acoustic models. As shown in Figure 1.2, the language model and/or pronunciation network may specify more than one allowable mapping between an utterance label and a sequence of lexical labels.

In general, the inventory of lexical labels must satisfy the following *decodability* criteria:

1. Each allowable utterance label must map to at least one sequence of lexical labels according to the rules of the lexicon; and

2. No two utterance labels (unless they are homonyms) map to the same sequence of lexical labels.

Furthermore, for good recognition performance, the set of lexical-acoustic units must be capable of discriminating among sequences of observation vectors associated with distinct lexical labels. To meet this requirement, the lexical-acoustic models should satisfy the following *discriminability* criteria:

1. The measurements $\boldsymbol{y}^T(t)$ made on each acoustic segment must capture sufficient information for such discrimination.

2. The model PDF of the sequence of observation vectors for a given label estimated from the training set should correspond closely to the true PDF of this sequence of vectors given the label. This requirement is related to the well-known pattern recognition result that a maximum a posteriori classifier achieves the highest theoretical performance (the Bayes rate) when the true PDF's for the class-specific measurement vectors are used [Duda 73].

The lexical-acoustic model associated with each lexical label is determined in the *training* process using a set of *training utterances*. The steps in this process are *transcription*, *acoustic segmentation*, and *acoustic model building*.

During transcription, each training utterance label is mapped to a sequence of lexical labels using the lexicon's rules. The mapping is in the opposite direction to that used in building utterance models out of lexical-acoustic units. Each training utterance is then transcribed so that each portion of its waveform is associated with a lexical label.[3] Formally, the transcription is a mapping between each lexical label and a time interval measured relative to the onset of the utterance. We will refer to the portion of the waveform associated with a particular lexical label as a *lexical region* and will say that the region is *occupied* by the label. More specifically, if the lexical label is a phone label, we will refer to the lexical region as a *phonetic region.*

Typically, automated procedures are used to transcribe waveforms into a time-aligned string of phonetic labels. The procedures are often based on "seed" transcriptions that have been determined manually by a transcriber. The transcriber typically chooses the label which he/she judges to best correspond to the acoustic quality of the waveform as heard or as represented in a spectrogram. Transcription boundaries tend to be placed at points that demarcate waveform regions possessing different acoustic qualities. Their placement is based on the transcriber's judgment.

During acoustic segmentation, the waveform is broken into acoustic segments according to a *segmentation algorithm* and a vector of acoustic measurements $\boldsymbol{x}^T(t)$ is associated with each segment where t indexes the measurement vectors. This measurement vector is often used directly as the observation vector so thaι $\boldsymbol{y}(t) = \boldsymbol{x}(t)$. However, it is sometimes useful to make a transformation $\boldsymbol{y}(t) = T(\boldsymbol{x}(t))$ and use the resultant observation vectors to train the models. In our work, we will often make use of linear transformations represented by a matrix multiplication $\boldsymbol{y}^T(t) = \boldsymbol{x}^T(t)\boldsymbol{T}$ and in these cases we will make a distinction between the measurement and observation vectors, the latter being used directly to train the state PDF's. We should point out, however, that in some cases we make several transformations between the original set of acoustic measurements made on each segment and the observation vector so that at times we will refer to a measurement vector that itself has been transformed from some other measurement vector. In all cases, our meaning

[3] For simplicity, we will not generalize the model to include stochastic transcription, in which more than one label may be associated to a portion of the waveform, with weights assigned to reflect strength of association.

should be clear from the context in which the terms are used.

Formally, each segment can thus be described by its observation vector along with a time interval measured from the start of the utterance. Thus, the segmentation process converts the waveform into a sequence of observation vectors, as stated above.

In most current recognition systems, the segmentation algorithm is simple. Each segment is a *frame* of the same duration, usually about ten milliseconds. The measurements made generally represent the segment's spectrum. They can be in the form of outputs of auditory [Seneff 86], simulated auditory [Davis 80], or linear predictive [Rabiner 83] models. The observation vector for a particular segment may also include measurements made on portions of the waveform outside the segment. For example, the dynamics of the spectrum in the vicinity of a segment may be represented as regression coefficients or as the difference in filter outputs measured on frames which precede and follow the segment [Furui 86]. We will refer to a system with fixed-duration acoustic segments as a *frame-based system*.

Figure 1.3 illustrates the process of training a model once transcription and acoustic segmentation have been accomplished. At this point, each lexical label in the training utterance can be associated with the sequence of acoustic segments (whose boundaries are denoted by dashed lines in the figure) occupying the same time interval. The set of sequences associated with a particular label over all the training utterances is used to estimate the parameters of the acoustic model associated with the label. This set constitutes the ***training data*** for the model. The arrows in the figure depict the associations between segments and labels. The resultant model is a statistical summary of the training data and is used to predict the sequence of observation vectors associated with the label that would be produced in a test utterance. Note that the underlying assumption made is that all information relevant to identifying the lexical label is included in the observation sequence associated with the label during transcription. However, this is not as much a limitation as

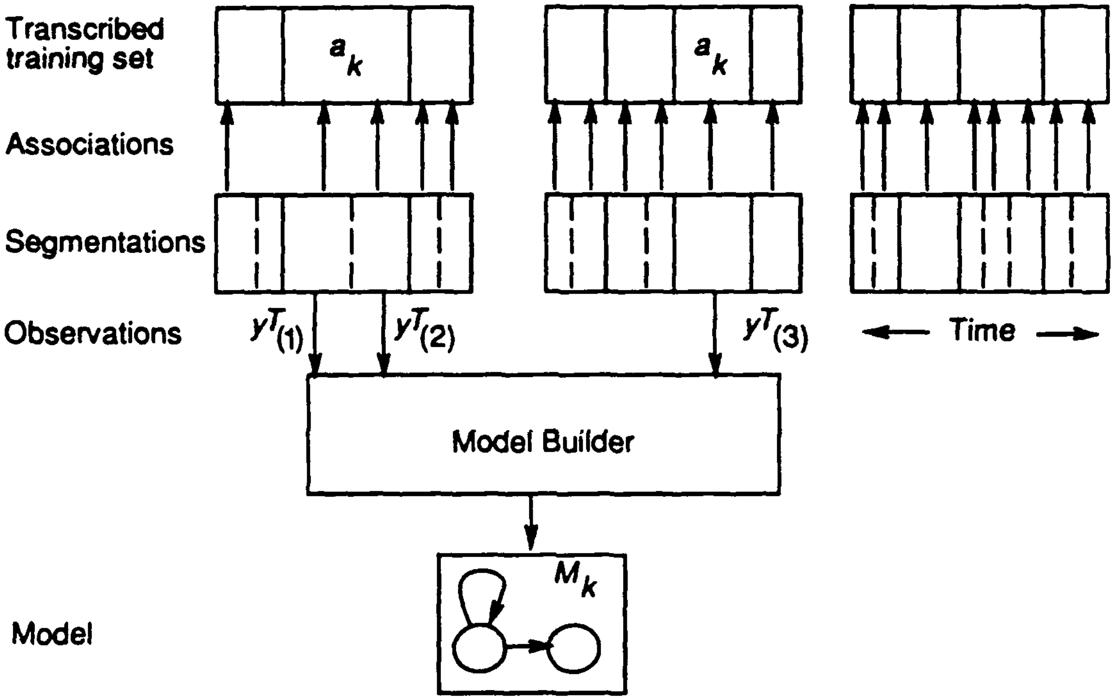


Figure 1.3: Training a lexical-acoustic model M_k. Regions denoted a_k occupied by lexical label a_k. Unlabelled regions occupied by other labels. Vertical lines in the segmentation are acoustic segment boundaries. Solid vertical lines are lexical region boundaries as well. Arrows depict associations between lexical regions and acoustic segments. Observation vector $y^T_{(i)}$ signifies the i^{th} training observation vector for model M_k.

it may seem since an observation vector made on a particular segment may include measurements based on portions of the waveform outside the segment boundaries. For example, measurements of spectral differences across one or both of the segment boundaries may be included.

It should be pointed out that this figure is an idealization in the sense that the transcribed label regions align exactly with segments, making the association between the two simple. In Figure 3.4, there is an illustration of an actual alignment between labels and segments produced by our segmentation algorithm.

In an HMM, an *alignment algorithm* such as the forward-backward algorithm [Jelinek 76] is used to associate acoustic segments with states in each lexical-acoustic model. For each state, ***transition probabilities*** are estimated which reflect the association between acoustic segment and state sequences (i.e., if there is a high transition probability from state r to state s, it is likely that an acoustic segment following a segment associated with r will be associated with s).

Simultaneously, parameters are estimated that represent the probability distribution function (PDF) of the observation vectors $\boldsymbol{y}^T(t)$ associated with each state. We will refer to this as the *state PDF*. The PDF is often represented as a multivariate Gaussian, for which a mean and a covariance are estimated. The vector quantization (VQ) representation is also used frequently [Rabiner 83, Bahl 81]. In this method, a number of prototype observation vectors is determined and each one is assigned an index. Each observation vector associated with a particular state is represented by the index of the prototype closest to it according to some distance metric. The state PDF is represented as a a histogram of relative frequencies of each index.

In the recognition process, the test utterance is segmented according to the same segmentation rule used in training and the same measurements made on the acoustic segments. Thus, the utterance can be represented as a sequence of observation vectors. The lexicon is used to combine lexical-acoustic models to search for the utterance acoustic model most likely to have produced the observation sequence. The corresponding utterance label is output by the recognizer. The search involves hypothesizing different utterance labels and computing the score for each of them. As mentioned above, an HMM system builds an utterance acoustic model out of a sequence of lexical-acoustic models. As in the training process, the scoring process involves aligning the sequence of observation vectors with the sequence of lexical-acoustic models. Thus, for each utterance hypothesis, there is an association between each lexical acoustic model and part of the utterance's observation sequence. We will refer to the corresponding mapping between each lexical label and the start and end times of the observation sequence as an ***hypothesized transcription***. As well, there is a likelihood that the observation sequence was produced by the lexical-acoustic model. We will refer to this likelihood as the ***acoustic match*** between the model and the observation sequence.

We should point out that the description in this section does not pertain to the class of statistical speech recognizers based on the stochastic segment model described by Ostendorf and Roukos [Ostendorf 89]. Two other examples of speech recognizers employing this model are the SUMMIT system [Zue 89a] and the system described in [Leung 92]. For these recognizers, the recognition process involves hypothesizing many possible segmentations and simultaneously choosing the model sequence and segmentation which yields the best recognition score. By contrast, in an HMM-based system, a segmentation is determined first and then the model sequence which yields the highest score for the observation sequence is found.

Formally, the two approaches are quite different. The score that an HMM assigns reflects the probability of observing a sequence of observation vectors. If each observation vector is of size q and there are T of them, the HMM score for a model sequence is an estimate of the probability that the sequence of models produced the observed point in a vector space of dimension qT. This is a well-formulated estimation problem. The same is not true in the stochastic segment model, for which different segment sequences must be scored against each other. Not only are the vector spaces upon which the probability estimation is made different, they are not of equal dimension. In practice, the varying dimensionality necessitates the use of an arbitrary normalization scheme, as in [Ostendorf 89]. While the stochastic segment model has been shown to yield results competitive with those of HMM's for some tasks [Digilakis 92], we prefer to use the HMM framework in the present work since, for the reasons just cited, it is more amenable to mathematical analysis.

Finally, we should point out that some of the assumptions made in the model we have described here can best be thought of as "engineering approximations" to reality. For example, it is simplistic to assume that the acoustic quality of each region of the waveform corresponds to just one phonetic label. In fact, the identity of surrounding labels and other factors affect acoustic quality. Approaches to dealing with this issue will be discussed further in Chapter 5. The problems with this assumption notwithstanding, to our knowledge no alternative to this process has been employed in building a statistical recognizer and it is beyond the scope of our work to develop such an alternative.

It has also been suggested that the idea of representing a word label as string of lexical labels is overly simplistic and that a more appropriate representation would involve a matrix of distinctive features [Stevens 90]. Again, it is beyond the scope of our work to explore such a representation.

Chapter 2

Speech Corpora and Signal Representation Techniques Used

In this chapter, we describe the speech corpora and signal representation techniques used throughout our work. Section 2.1 provides a brief description of the VOYAGER corpus and outlines why it was chosen. Section 2.2 describes the VOYAGER and TIMIT corpora in more detail, concentrating on signal processing and phonetic transcription. Finally, Section 2.3 describes the hair-cell envelope and mel-frequency spectral representations, both of which are used in the present research. That section also describes principal component analysis, a scheme used to reduce the dimensionality of the spectral representations.

2.1 The VOYAGER Corpus

VOYAGER is a system which responds to natural-language queries about the geography of Cambridge, MA as well as to queries about private and public establishments within Cambridge, such as post offices, libraries, and restaurants. The speech corpus was collected from fifty male and fifty female speakers interacting with the system for about twenty minutes each. Orthographic transcriptions of each sentence spoken spontaneously during the interaction period were subsequently presented to the speaker to be read. Thus, each sentence is represented by two utterances, one spontaneous and one read. We use only the read sentences in our work.

We chose this corpus because it includes relatively large numbers of particular content words, including proper nouns such as "Baybank", "Harvard", and "MIT" In Chapter 5, we examine the effect of word model type on performance in relation to whether the modelled word is a content or function word. Thus, the VOYAGER corpus is useful for our purposes.

2.2 Details of the TIMIT and VOYAGER Corpora

The TIMIT and VOYAGER corpora were recorded under similar conditions, using a Sennheiser close-talking noise-cancelling microphone, and were digitized with DSC 240 audio control boxes [Zue 89c, Fisher 86]. TIMIT was initially lowpass filtered to 8 kHz, sampled at 20 kHz and downsampled to 16 kHz while VOYAGER was filtered at 6.4 kHz and sampled at 16 kHz Since the spectral representations described below use bandpass filters whose center frequencies are all below 6.4 kHz, we felt that the difference in cutoff frequencies would not pose a problem in combining training data from the two corpora and using VOYAGER test data. Early on in our work, we verified that the corpora were compatible by performing an experiment comparing word spotting performance using VOYAGER training data alone and both VOYAGER and TIMIT training data. Since use of the extra data from TIMIT improved performance we decided to retain the two corpora in the training set.

The corpora were transcribed using different methods, however. The TIMIT utterances were first phonetically transcribed by a phonetician and then aligned with the CASPAR alignment system [Leung 84]. Those of VOYAGER were first automatically transcribed and aligned using the SUMMIT recognizer and were then checked and modified manually. In this automatic transcription process, each word in each utterance was represented as a network of phonetic symbols.

A set of phonological rules [Zue 90a] was used to produce the networks. Consequently, each word was represented by more than one sequence of phonetic labels. The SUMMIT recognizer was used to find the aligned sequence of phonetic labels that produced the highest acoustic score on each utterance being transcribed. Even though the resulting transcriptions were checked manually, there is evidence that there was a tendency among the transcribers to preserve the automatic transcriptions unless there were gross errors. Thus, most of the variability among the phonetic strings used to represent each word in the VOYAGER corpus is probably due to the phonological rules and SUMMIT system. This is an issue because the transcriptions are used in our work to train both subword and word models and to evaluate phonetic recognition performance. Where it is pertinent, we will discuss the effect of this transcription technique on our results.

For our work on word spotting, we required time-aligned orthographic transcriptions as well. For the TIMIT corpus, these transcriptions were provided for each utterance. For the VOYAGER corpus, the transcriptions were derived from the VOYAGER time-aligned phonetic transcriptions using the AAT program [Kassel 86]. Given phonetic and orthographic transcriptions for an utterance and a dictionary associating each word in the training data vocabulary with allowable phonetic spellings, the program aligns the two transcriptions. The alignment is used in conjunction with the utterance's time-aligned phonetic transcription to produce a time-aligned orthographic transcription.

In both corpora, silence regions before and after each utterance were retained and labelled with a special transcription symbol. Rather then employ and endpoint detector or build a model for these regions, we removed the silences before further processing according to the endpoints indicated in the transcription. We also removed from the beginning and end of the utterances any non-speech events such as lip smacks and pauses that were identified in the transcription.

Only data from male speakers were used for training and testing purposes.

We made this decision because there is evidence that gender-specific acoustic models outperform models trained from both sexes [Bush 87, Kubala 91]. Rather then building two sets of models or addressing issues of speaker adaptation or normalization, we decided to build models for speakers of just one gender. Male speakers were chosen because they form 70% of speakers in the TIMIT database and so provided more training data. We will specify in greater detail the training and test sets used for each experiment where we describe the experiment.

2.3 Spectral Representations

We used two different spectral representations in our work, the hair-cell envelopes and the mel-frequency spectrum. Before computing either of these representations, each waveform's energy was normalized by scaling the waveform so that the maximum absolute sample value over the entire utterance was constant for all waveforms.

2.3.1 The Hair-Cell Envelope Representation

The hair-cell envelope (HCE) representation is derived from the Seneff Auditory Model [Seneff 86]. It consists of a set of coefficients, each representing the firing rate of hair cells tuned to a particular frequency band. The frequency bands used correspond to psychophysical critical bands [Zwicker 61]. For the 0-6.4 kHz bandwidth, there are 40 coefficients in the representation. The hair-cell response in each band is a monotonically non-decreasing function of the energy in the band. The function is non-linear and, in particular, the response is set to 0 below a particular energy threshold and is clipped to a maximum value for energies above a threshold. The set of firing rates is sampled every 5 ms. The representation has been used in several phonetic classification and speech recognition systems [Glass 88, Leung 89, Meng 90, Leung 92, Niyogi 91, Zue 89a, Sorensen 89].

In our work, we used the hair-cell envelopes in the segmenter and for the first set of phonetic recognition and word spotting experiments we performed. However, for reasons to be discussed in Chapter 4, we used the mel-frequency spectral representation in subsequent experiments.

2.3.2 The Mel-Frequency Spectral Representation

The mel-frequency spectrum [Davis 80, Meng 90] is also a representation of the energy within each of a set of frequency bands. In our implementation, it is sampled every 5 ms and consists of 40 coefficients. To compute it, a short-time Fourier transform is computed on a pre-emphasized waveform every 5 ms using a 25 ms Hamming analysis window. Within each of 40 frequency bands, the magnitudes of the Fourier coefficients are multiplied by a triangular window and summed. We will refer to the triangular windows as filters throughout the thesis. The mel-frequency spectral coefficient (MFSC) associated with a given filter is the logarithm of the sum of the windowed Fourier coefficient magnitudes in that band. The filter centers are linearly spaced below 1 kHz and logarithmically spaced above 1 kHz, i.e., they are placed at equal intervals along the mel scale and have a bandwidth proportional to their spacing. The spacing is designed to model the psychophysical critical bands. Mel-frequency representations have been used in a number of speech recognition and phonetic classification systems including those described in [Gupta 87, Chow 87, Lee 88, Ostendorf 89, Cohen 90, Paul 91, Digilakis 92].

For most speech recognition work, including that described in the aforementioned references, the coefficients are linearly transformed into mel-frequency cepstral coefficients using a cosine transformation [Davis 80]. Only the lower-order coefficients (i.e., those produced by the lowest-quefrency[1] cosines) are retained. The main motivation for this transformation is that the lower-order coefficients are related to the coarser features of the spectral shape which are believed to be the most useful for phonetic discrimination. Thus, the transfor-

[1] quefrency is the cepstral-domain analog of frequency

mation serves as a dimensionality reduction technique. Reducing dimensionality is desirable because it reduces the computation required in computing acoustic scores and the storage required for a system's lexical-acoustic models. Additionally, given limited training data, reducing dimensionality can lead to improved recognition performance [Duda 73].

2.3.3 Principal Component Analysis

We have chosen to accomplish dimensionality reduction using principal component analysis [Johnson 88]. We used this method to reduce the dimension of both the hair-cell and mel-frequency representations. Given a set of measurement vectors each of length p, the method is used to find the linear combination of measurements which has the highest variance. That linear combination is the first principal component. The second principal component is the linear combination orthogonal to the first component that has the highest variance, and so forth. Thus, the method produces p principal components in decreasing order of variance. This technique has often been used to reduce the dimensionality of spectral representations. See, for example, [Plomp 67, Pols 69, Klein 70, Pols 73, Bocchieri 86, Glass 88].

Principal components are computed by performing an eigenanalysis of the sample covariance or sample correlation matrix. The resulting variables are referred to as covariance and correlation principal components, respectively. When the variances of the measurements differ widely, covariance principal components tend to lie in the directions of the measurements with largest variance, regardless of the structure of the data. In such a case, correlation analysis yields more useful results. In fact the results of correlation analysis are the same as those that would be obtained if the original measurements were first scaled so that they each had the same variance and were then subject to covariance analysis. Put another way, if the measurements each had the same variance to start out with, covariance and correlation principal components would yield the same results. Thus, correlation principal components are more general. For this reason, we chose to use them even though neither the hair-cell-envelope or mel-frequency-spectral coefficients have widely different variances. Nonetheless, the discussion that follows pertains to covariance principal components since this method is easier to describe.

The first q principal components can be described in terms of the proportion of variance in the measurement set for which they account. The precise meaning of this quantity can be explained as follows: Let v_j be the variance of the jth measurement and e_k be the variance of the kth principal component. Let $V = \sum_{j=1}^{p} v_j$ be the total variance of the measurement set. It can be shown that $V = \sum_{k=1}^{p} e_k$. The quantity $\sum_{k=1}^{q} e_k/V$ is referred to as the proportion of variance accounted for by the first q principal components.

Thus, the main purpose of principal component analysis is the same as that of the cosine transformation: to rank the transformed variables by their ability to capture coarse spectral shape features so as to reduce dimensionality. However, the analysis is guaranteed to accomplish this goal for any data set while the cosine transformation is not. See Glass [Glass 88] for a comparative description of the two techniques.

In fact, using a least-square error criterion, principal component analysis is the optimal dimensionality reduction technique. To be specific, assume an original set of n data points, each represented by a vector of p measurements. Let $\boldsymbol{X}$ be an $n \times p$ data matrix such that x_{ij} is the value of measurement j on data point i. Assume that each point is linearly transformed into a vector whose length is q, $q < p$ and let y_{ik} be the value of the kth transformed variable on data point i. This can be represented by the matrix multiplication $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{A}$ where $\boldsymbol{A}$ is a $p \times q$ transformation matrix and $\boldsymbol{Y}$ is an $n \times q$ data matrix. Since $q < p$, this represents a reduction in dimensionality. A reasonable way to gauge the amount of information about the original measurement set $\boldsymbol{X}$ that is retained by this process is to measure the accuracy to which it can be approximated from the transformed variable set $\boldsymbol{Y}$ by *back-transforming* $\boldsymbol{Y}$ to the original vector space. In particular, let $\hat{\boldsymbol{X}} = \boldsymbol{Y}\boldsymbol{B}$ be an $n \times p$ matrix representing the minimum squared error approximation of $\boldsymbol{X}$ from $\boldsymbol{Y}$. Let $\hat{x}_{ij}$ be the value of measurement j on data point i in $\hat{X}$. Thus, $\boldsymbol{B}$ is a $q \times p$ matrix determined so as to minimize the quantity

$$E = \sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij} - \hat{x}_{ij})^2.$$

over all $q \times p$ matrices. It can be shown [Johnson 88, p. 367] that the set of q transformed variables that achieves the minimum E is the set of the first q principal components. Furthermore, in that case E is proportional to the variance not accounted for in the first q principal components. Thus, the proportion of variance accounted for can be used to determine the accuracy of the approximation.

Figure 2.1 illustrates for two sample spectra and for different values of q the approximation obtained by back-transforming the principal components as discussed above. It can be shown [Johnson 88, p. 367] that the back-transformed estimate of the vector of spectral coefficients is $\boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{\mu}_x[\boldsymbol{I} - \boldsymbol{A}^T\boldsymbol{A}]$ where $\boldsymbol{A}$ is the principal component transformation matrix, $\boldsymbol{y}$ is the vector of principal components, $\boldsymbol{\mu}_x$ is an estimate of the mean spectral coefficient vector and $\boldsymbol{I}$ is the identity matrix. The figure displays this estimate for a back-transformation to the mel-frequency spectral coefficients. Filter outputs are plotted at their center frequencies which are spaced in the mel domain along the horizontal axis. The first spectrum was extracted from an /s/ and the second from an /ɑ/. In each figure, the solid line represents the original spectrum and the dotted line represents the approximation. As q is increased, the approximations to the original spectra are improved. From these figures, it appears that coarse spectral features are preserved while fine ones are not.

Note that principal component analysis is an optimal dimensionality reduction technique only in the sense of minimum squared error. It is possible that the fine spectral information that is lost in the process is useful for discriminating lexical labels. We discuss this point in more depth in Section 4.7, where we discuss multiple discriminant analysis, a dimensionality reduction

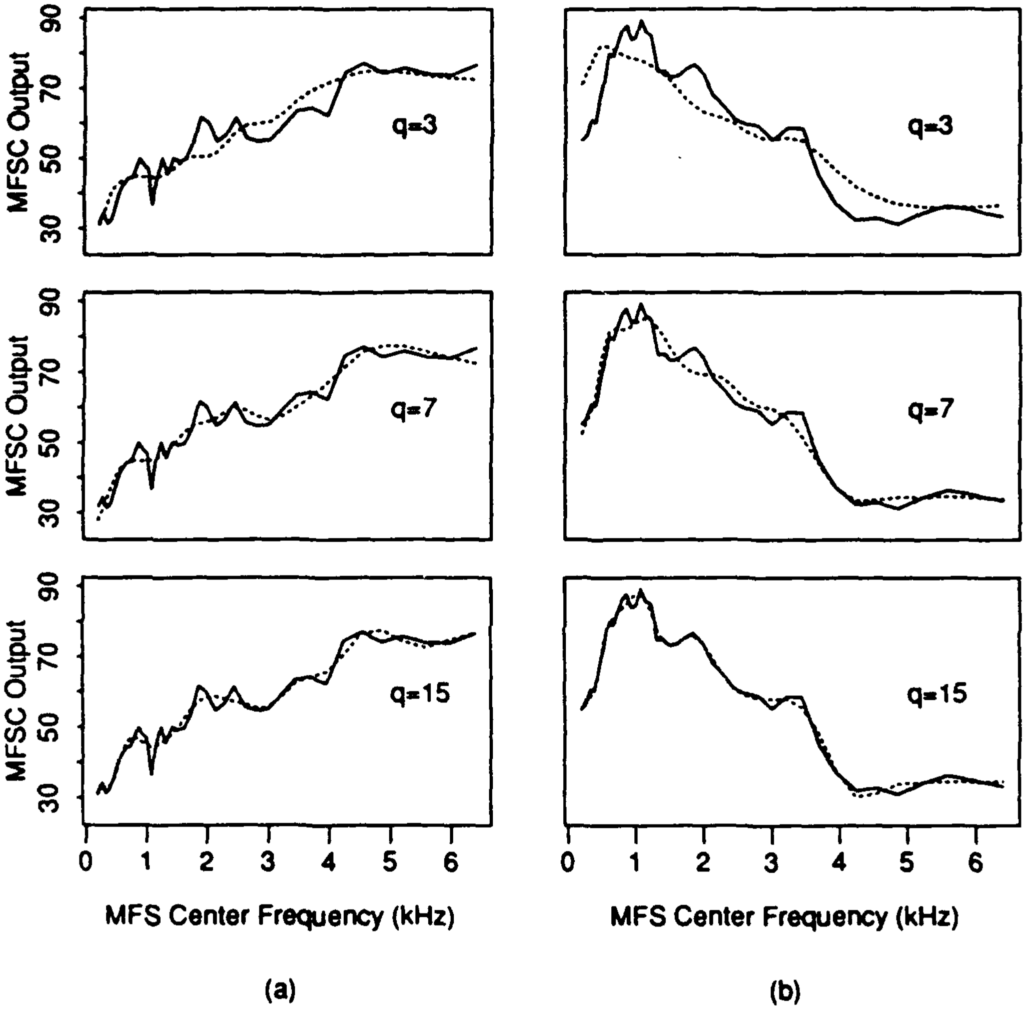


Figure 2.1: Effect of principal component dimensionality reduction. The mel-frequency spectral representation is used. Filter outputs are plotted at their center frequencies which are spaced in the mel domain along the horizontal axis. Each solid line represents the original spectrum and each dotted line the least-square approximation. Approximations shown for 3, 7, and 15 principal components representing 86%, 94% and 98% of variance, respectively. (a) Spectrum extracted from /s/. (b) spectrum extracted from /ɑ/.

technique designed to preserve information useful for discrimination. The distinction between the two techniques was noted in [Pols 73] and discussed in depth in [Brown 87]. In our work with MFSC's, we represent them with fifteen principal components. These account for 98% of the variance so the loss of information is probably not a major consideration. The hair-cell envelope representation is represented by seven principal components accounting for 90% of variance so the loss of information may be more of a consideration.

Chapter 3

The Segment-Based HMM

Our system differs from frame-based systems usually used in HMM recognizers in that the role of frames is played by acoustic segments of varying duration that are generated by an algorithm that places boundaries at points of large spectral change. In particular, these points are intended to define boundaries between regions associated with different lexical labels, using the terminology of Section 1.3. In this chapter, we justify our use of such a scheme, describe the segmentation algorithm, characterize its performance, and present our solutions to some problems that occur in using this algorithm for speech recognition. Finally, we discuss how subword models are built within the segment-based HMM framework. These models are used in the phonetic recognition and word spotting experiments discussed in Chapters 4 and 5.

3.1 Previous Work on Phonetic and Variable-Duration Segments

The idea of making measurements on phonetic regions is not new. A few examples of this include work done on recognition of vowels [Leung 89], fricatives [Key 87], and nasals and nasalized vowels [Glass 84]. In these cases, the regions were determined manually.

Automatically determined segments have been used for isolated letter recognition [Cole 86], connected digit recognition [Bocchieri 86, Bush 87] and in the stochastic segment speech recognizers mentioned in Section 1.3 [Ostendorf 89, Zue 89a, Leung 92]. As we discussed in that section, these systems differ from the HMM-based systems in that they perform segmentation and recognition jointly, and thus are formally quite different from the system used in the present work.

In fact, to our knowledge, only one hidden Markov model system using variable-duration segments has been built. In [LeMaire 89] a segmenter based on sequential hypothesis testing is used to determine points of large spectral change at which segment boundaries are placed. The segmenter is described in [André-Obrecht 88]. Preliminary results for digit recognition using the recognizer were promising.

A related approach, employing a variable frame rate, is described in [Peeling 91]. The measurements in this system were the same as in a frame-based system but the frame sampling rate varied as a function of the local rate of spectral change. The authors reported that the variable frame rate system outperformed the fixed-rate one.

Finally, we should point out that early work by IBM [Jelinek 76] modelled the segment rather then the frame as the unit upon which acoustic scores were computed. However, the hidden Markov models used in this work were used to model a channel with discrete input and output symbols rather then measurement PDF's. Thus, this system is not well-described by the speech recognition model of Section 1.3 that we use in our work.

An approach similar to that of [Jelinek 76] is described in [Meisel 91], using a segmenter based on automatic neural networks. That paper reported good results for a phonetic recognition task.

3.2 The Ideal Segmenter

To motivate the segmentation algorithm, we will introduce the concept of an *ideal segmenter*. This concept is based on the assumptions that each word in the vocabulary can be represented as a distinct sequence of lexical labels and that an *ideal transcription* (i.e, a mapping between waveform regions and label sequences) can be determined such that the acoustic quality of the waveform region associated with each label is dependent solely on the label's identity. Looked at another way, the waveform region associated with each label would provide all the information relevant to identifying the label. The assumption that such a transcription exists was introduced in Section 1.3. We will also assume that a manual transcription comes closest to an ideal transcription. An ideal segmenter would automatically produce a segmentation that corresponds to an utterance's ideal transcription.

As we discussed in Section 1.3, the point of training models is to estimate a PDF for the sequence of observation vectors associated with each lexical label. With an ideal segmenter, this would be reduced to the problem of estimating the PDF for the single observation vector associated with each occurrence of a lexical label. Since, by assumption, this observation vector yields all the information relevant to identifying the label, such a system would provide the best possible discrimination of labels for a given lexical label inventory and choice of observation measurements. In turn, since each word in the recognizer's vocabulary can be represented as a distinct sequence of lexical labels, such a system would provide the best possible word recognition given these choices.

Evidence that an ideal segmenter would lead to better recognition performance than a non-ideal segmenter is cited in [Ostendorf 89]. The recognition rate of phonemic labels for the stochastic segment recognizer discussed in that paper was higher when manual transcription was used to determine the segment boundaries of the utterances to be recognized than when an automatic segmentation was used. In the automatic segmentation, segment boundaries were placed less consistently for different tokens associated with the same transcription label. Presumably, the automatic segmentation did not approximate an ideal segmentation to the same extent as did the manual segmentation and this caused a deterioration in recognition performance.

For the remainder of the chapter, we will discuss an ideal segmenter based on lexical labels consisting of phones. As introduced in Section 1.3, the term *phonetic region* will refer to the portion of the waveform associated with a particular phone label. The segments produced by the segmenter will be referred to as *phonetic segments.*

3.3 Rationale for Using Acoustic Segmentation

In the previous section, we discussed in broad terms the rationale for using measurements made on phonetic segments as observation vectors. In this section, we provide more detailed arguments for attempting to produce a phonetic segmentation in a speech recognizer.

The first justification is that there is evidence that points of large spectral change may be particularly rich in phonetic information about the identity of certain consonants [Stevens 85, Stevens 86] so that measurements made at these points will be found to be useful. The segmenter is designed to identify these points so that making these measurements is convenient within the segment-based HMM framewo:k. In Chapter 4 we show that such measurements can in fact be used to discriminate among stop consonants.

Use of phonetic segments may help in dealing with the problem of statistical dependence among frames in a frame-based system. Frame-based systems compute the probability of observing a sequence of frames under the assumption that adjacent observation vectors are statistically independent given the current Markov model state. This has been shown to be a faulty assumption, especially for long vowels for which adjacent frames are highly correlated. For a good discussion of the problem, see [Brown 87]. Both Brown [Brown 87] and Kenny [Kenny 90] have suggested solutions to the problem within the HMM framework, obtaining limited success. Digilakis [Digilakis 92] obtained some success in dealing the problem within the stochastic segment framework.

The consequence of the correla: on is that observation probabilities for frames associated with long vowels tend to dominate the score computed for the full utterance. Thus, for example, in an isolated word recognizer a word like "zoo" may be recognized as "so" in the following scenario: The average spectrum for the /u/ matches slightly better to the acoustic model for /o/ than to the /u/ model. At the same time, the uttered /z/ might match much better to the correct model than it does to the second-best matching model, /s/. Because the vocalic frames dominate the computed utterance likelihood, the recognizer will output a word ending in /o/. Since there is no word whose phonetic transcription is /zo/, it is likely that the recognized word would be "so." In a system in which the lexical-acoustic models account for interframe correlation, the poor match of the uttered /z/ to the /s/ model would overwhelm the relatively small tendency of the recognizer to output /o/ and the recognizer would respond correctly.

To overcome this problem, correlations between nearby spectra must be modelled and/or adjacent acoustic segments should be as statistically independent as possible. Phonetic segment systems can satisfy both these criteria better than do frame-based systems. Theoretically, models for phonetic segment observation vectors which include short-term spectra can model correlations between nearby spectra, although detail in the correlation structure will be limited by the amount of training data available. The problem of statistical dependence among frames associated with the same phone is due in large part to the fact that spectra tend to change more slowly withir phonetic regions than across phonetic boundaries. Because there is relatively little change from one frame to the next, one frame's spectrum can be well predicted given the previous frame's. A system based on phonetic segments reduces this intersegment dependence by tending to place intersegment boundaries at boundaries between phonetic regions. Thus, phonetic segments should partially alleviate the problem of poorly modelled statistical dependence among segments.

Finally, an HMM recognizer based on phonetic segments can be computationally cheaper than one based on frames. In continuous speech systems, much of the computation is devoted to searching for the best sequence of lexical labels among all those sequences allowed in the lexicon. The computation time required for the search is linear in the number of acoustic segments in the utterance [Rabiner 83]. Because phonetic regions tend to be longer than the typically used 10 ms frame, each utterance has fewer of them, leading to computational savings. In our implementation, for example, the mean segment length is about 50 ms, so the segment rate is only 1/5 the typical frame rate in a frame-based system. This advantage must be balanced against the computation required for the segmentation itself, which is not a factor in a frame-based system. We make a rough analysis of the computation required for segmentation in Section 4.8.3 and show that it is small compared to the time required for phonetic recognition. We should point out that the same is not true of most stochastic segment models, unless they take steps to greatly constrain the search for the optimal segmentation [Leung 90, Digilakis 92]. A rough analysis in the same section of the requirements for the stochastic segment systems reported in [Leung 90, Digilakis 92] shows large differences between the two approaches. The extra computation required for these systems compared to ours can be traced to the fact that these systems use a large number of phone models for both segmentation and recognition while we partition the two tasks and use the phone models only for recognition.

3.4 Segmenter Description

3.4.1 Overview of Algorithm

The goal of the segmenter is to produce an output as close to the phonetic segmentation as possible, i.e., to generate segments that are aligned as closely as possible to the phonetic transcription. Figure 3.1 depicts the training process.

The segmenter training set consists of utterances which have been phonet-

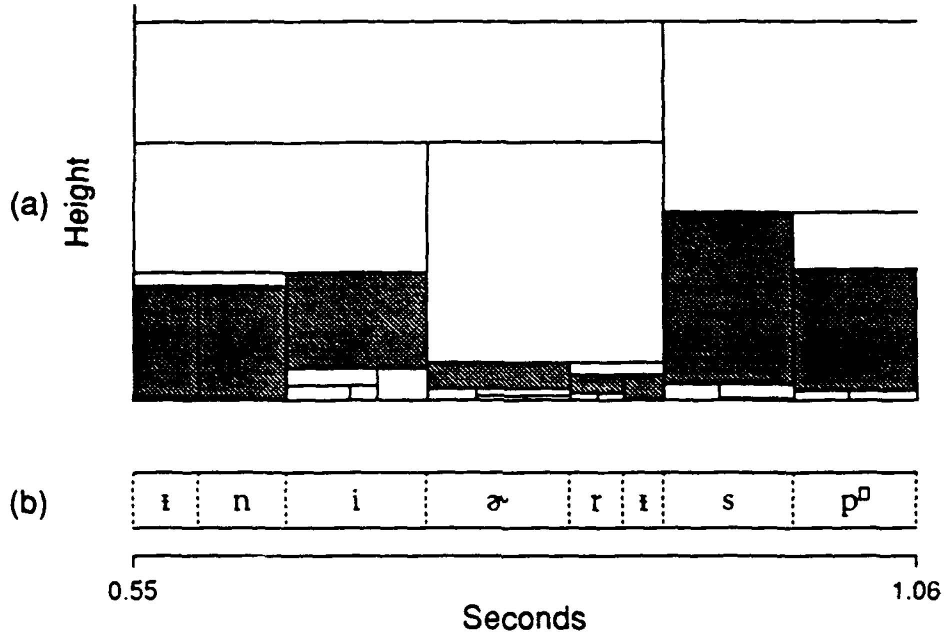


Figure 3.1: (a) Multi-level segmentation (MLS) and (b) phonetic transcription. Displayed times measured from beginning of utterance. The shaded regions in the MLS are good segments. Those above them are merge segments, those below are split segments. Good segments are determined using dynamic programming to find the sequence of segment boundaries in the MLS that most closely match the transcription boundaries. The MLS height is defined in Section 3.4. A total of 57 phone labels were used for transcription.

ically transcribed manually. The transcribed phone labels include those that correspond to phonemes as well as to acoustically distinct allophones such as alveolar flaps and glottal stops. The inventory also includes several subphoneme labels such as stop closures and releases whose acoustic properties are quite distinct from other labels. These were adopted in order to satisfy the "separation" criterion among labels discussed in Section 1.3. A total of 57 labels was used. As discussed in Section 1.3, a higher degree of separation in the acoustic space between segments associated with different labels should lead to higher recognition accuracy.

To train the segmenter, a multi-level segmentation (MLS) [Glass 88] is first computed for each utterance in the segmenter's training set. The MLS organizes the utterance into a network of hypothesized acoustic segments. We classify the set of acoustic segments produced by the MLS on a segmenter training utterance into three types. "Good" segments are those whose time boundaries match most closely to those of the transcription. These are found using dynamic programming as described in [Glass 88]. The good segments are displayed as shaded regions in Figure 3.1. We treat the sequence of good segments as the closest approximation to a phonetic segmentation attainable given the set of MLS segment hypotheses. "Split" segments are those which are subsumed by good segments. These appear below the good segments in Figure 3.1. "Merge" segments, in turn, subsume the good segments and appear above them in Figure 3.1. Measurements useful for distinguishing among the three segment types are made on each segment and their PDF's are modelled by the training algorithm.

In the segmentation process, the same measurements are made on each MLS segment and the probability that the segment belongs to each one of the three segment types is estimated given the measurements. The segmenter finds the sequence of adjacent segments in the MLS that minimizes a cost function of the probabilities. The process is described in more detail in the next three subsections. In Sections 3.4.2 and 3.4.3, we describe the training and segmentation processes and in Section 3.4.4 we present the results of experiments used to determine the segmenter measurement set and to set certain system parameters.

3.4.2 Training

The training algorithm is based on vector quantization (VQ) and operates as follows: First, all measurements used to classify the segments as merged, split or good are scaled so as to have equal sample standard deviations. Measurement vectors of all three segment types are assigned to K clusters using a variant of the Linde-Buzo-Gray top-down clustering procedure described in [Lee 88] and introduced in [Linde 80] in which a cluster originally including all the training vectors is repeatedly split into two according to a criterion that maximizes the ratio of between-cluster and within-cluster spread. We refer to K as the VQ codebook size. For each cluster k the probabilities P_{kg}, P_{ks} and P_{km} that the cluster is associated with a good, split or merged segment are estimated. To describe how these estimates are made, we must first define some notation. Let D_{jk} be the squared Euclidean distance between segment j's measurement vector and cluster center k. Thus,

$$D_{jk} = \sum_{i=1}^{p}(C_{ki} - x_{ji})^2, \qquad 1 \leq k \leq K, \qquad 1 \leq j \leq N \tag{3.1}$$

where C_{ki} is the value of the ith measurement at the cluster center, x_{ji} is the value of the ith measurement made on the segment, p is the number of measurements, and N is the number of training vectors.

For each segment, define an *association* function

$$f_{jk} = \frac{\exp(-D_{jk})}{\sum_{i=1}^{K}\exp(-D_{ik})}, \qquad 1 \leq k \leq K, \qquad 1 \leq j \leq N \tag{3.2}$$

that measures the degree to which segment j is associated with each cluster. Clusters whose centers are close to that of the segment measurement vector are assigned high association values compared to those whose centers are far and the values are normalized so that their sum over all clusters is unity.

Let $\mathcal{G}$ be the set of indices of good segments in the segmenter training set. For each cluster k, a measure of F_{kg}, the number of good segments associated with it is

$$F_{kg} = \sum_{j \in \mathcal{G}} f_{jk}, \qquad 1 \leq k \leq K. \tag{3.3}$$

Analogously, the values F_{ks} and F_{km} can be defined for the split and merged segments, respectively. Finally, for each cluster, the values P_{kg}, P_{ks} and P_{km} are set according to:

$$P_{kx} = \frac{F_{kx}}{F_{ks} + F_{kg} + F_{km}}, \qquad x \in g, s, m \tag{3.4}$$

These values are used in the segmentation process described in the next section.

We have chosen to characterize the segment measurement PDF's using vector quantization because the method makes few assumptions about the form of the PDF's and is thus more more general than a Gaussian PDF model, for example. This quality is particularly important for the segmentation task because good segments correspond to phone labels of all types and the shape of their PDF's is likely quite complicated. Similarly, split and merge segments are also likely to have complicated measurement vector PDF's.

3.4.3 Segmentation

To describe the segmentation algorithm, we must first briefly outline the structure of the MLS. A more complete description is included in [Glass 88]. The MLS is organized hierarchically. The lowest level of the hierarchy consists of "seed regions". These are represented in Figure 3.1 by the bottommost segments. Seed regions are determined in a manner such that boundaries between them tend to represent points of large spectral change relative to the spectral change within them.

To build the MLS, the pair of seed regions which are most like each other acoustically according to a spectral distance metric are merged to form a new segment which is said to be the "parent" of the pair. The MLS used in our work is computed using the algorithm described in [Glass 88] but uses a different spectral distance metric for merging regions that is described in [Zue 90b].

The starting and ending times of the parent segment are the starting and ending times of the left and right "children", respectively. The spectral distance between the children is referred to as the "height" of the parent segment. The height is represented along the figure's vertical axis. Once the first pair of segments has been merged, the process continues, merging the two previously unmerged adjacent segments most acoustically alike until the entire utterance is represented by one topmost segment. Note that once two children are merged neither of them are considered for future merges. Thus, each segment (except for the topmost one) has exactly one parent and each parent has exactly two children, which are related to each other as "siblings." Finally, note that Figure 3.1 represents only part of the complete MLS computed for an utterance. Thus, the MLS's topmost segments are not displayed.

We use the MLS as the first step in the segmentation algorithm because it has been shown that a path can usually be found through it consisting of segments that align closely with the manual transcription [Glass 88]. As we stated in Section 3.2, the goal of the segmenter is to find a sequence of such segments. The MLS provides a structure that constrains the search for this sequence.

The segmentation algorithm operates by ascending from the seed regions upwards in the MLS, starting with the pair of seed regions at the beginning of the utterance, until it determines that the likelihood that the parent of a sibling pair is a merge segment exceeds some threshold. When this occurs, its children are both labelled good segments and appended to the sequence of good segments already determined. The algorithm proceeds in this fashion from left to right along the utterance until the end.

For each segment j, the odds ζ_j that it is a merge rather then a good segment is used in the algorithm and is defined as

$$\zeta_j = Q_{jm}/Q_{jg} \tag{3.5}$$

where Q_{jx} denotes the estimated probability that the segment belongs to class $x, x \in \{g, s, m\}$. It is estimated as

$$Q_{jx} = \frac{\sum_{k=1}^{K} P_{kx} \exp(-D_{jk})}{\sum_{k=1}^{K} \exp(-D_{jk})} \tag{3.6}$$

where the D_{jk} and P_{kx} are computed as described in Equations 3.1-3.4.

The algorithm is described in Table 3.1. It evaluates ζ_j of the parent of a segment instead of the segment itself to enforce a non-overlapping segmentation. If we did not adopt this strategy then a segment's parent could be considered twice for inclusion in the segmentation, once when ascending the MLS from the left child and once from the right child. This could lead to a child and its parent, which overlaps it, both being included in the segmentation.

3.4.4 Determination of Measurement Set and System Parameters

We experimented with several measurement sets and codebook sizes in developing the algorithm. For these experiments, the segmenter was trained using data from forty male speakers from the TIMIT corpus and tested with a disjoint set of ten male speakers from the same corpus. As described in Section 3.4.1, transcriptions of the utterances were used to label the segments in the training and test MLS's as good, split or merged. These labels were used for training the segmenter and for evaluating segmenter performance.

To choose a measurement set, Λ was set to 1. With this choice, the algorithm produces the topmost segments in the MLS that are more likely to be good than merged. Thus, this choice favors good segments over merged ones. At the same time, it favors good segments over split ones because the topmost segments likely to be good are used. We reasoned that this choice would

Definitions:

b: the starting time of the current seed region,
v_b: the seed region that begins at time b,
e_j: the ending time of segment j,
S: the current sequence of segments in the segmentation,
T: the ending time of the utterance being segmented,
j: the current segment,
A_j: the parent of segment j,
L_j: the left child of segment j,
R_j: the right child of segment j,
Λ: threshold that controls the number of segments produced
ζ_j: merge/good likelihood (see Eq. 3.5)

Algorithm:

```
             $b = 0$;                              (Start at time 0)
             $S = \{\}$;                           (Start with no segments)
StopRule:    if $b = T$ stop;
             $j = A_{v_b}$;                        (Ascend MLS)
MergeTest:   if $\zeta_j > \Lambda$ then
                 $S = SL_jR_j$;                    (Append to segmentation)
                 $b = e_j$;                        (Move to beginning
                                                     of segment following $j$)
                 goto StopRule;
             else
                 $j = A_j$, goto MergeTest;        (Ascend MLS)
```

Table 3.1: Segmentation algorithm.

maximize the number of good segments produced and that it would lead the segmenter to produce about one segment per transcription label in the test set since there would be no bias towards either split or merged segments.

The number of test utterance segments produced by the segmenter that were labelled as good segments in the MLS was used as a criterion for evaluating segmenter performance as a function of measurement set. This criterion was used because it is a measure of how close the algorithm comes to producing the ideal segmentation, in which all segments are good ones.

Figure 3.2 depicts the evaluation process. In the figure, an MLS, segmenter output, and phonetic transcription are shown for part of an utterance. Cross hatchings in the "northeast to southwest" direction indicate good segments in the MLS. These are the segments that match most closely to the phonetic transcription. Cross hatchings in the "northwest to southeast" direction indicate segments in the MLS that were found by the segmenter. Good segments found by the segmenter are cross-hatched with both patterns. Thus, for this example, there were five good segments, out of which three were found by the algorithm.

Our strategy for choosing a measurement set is based on the assumption that merge segments will tend to have the greatest within-segment spectral change and be the longest in duration while split segments will have the opposite attributes. Good segments should fall in between. Thus, the measurement set should be sensitive to within-segment spectral change and duration. We used the segment's height in the MLS in our measurement set to reflect spectral change. As we stated in Section 3.4.3, the segment's height is a gauge of spectral distance between the left and right segment children. The boundary between children tends to be a point of relatively large spectral change. Thus, the segment height should be sensitive to a large spectral change within the segment. In particular, if the segment is a merge segment whose left and right children have very different acoustic characteristics (e.g., a stop burst on the left and a vowel on the right), the segment height should be quite large. To

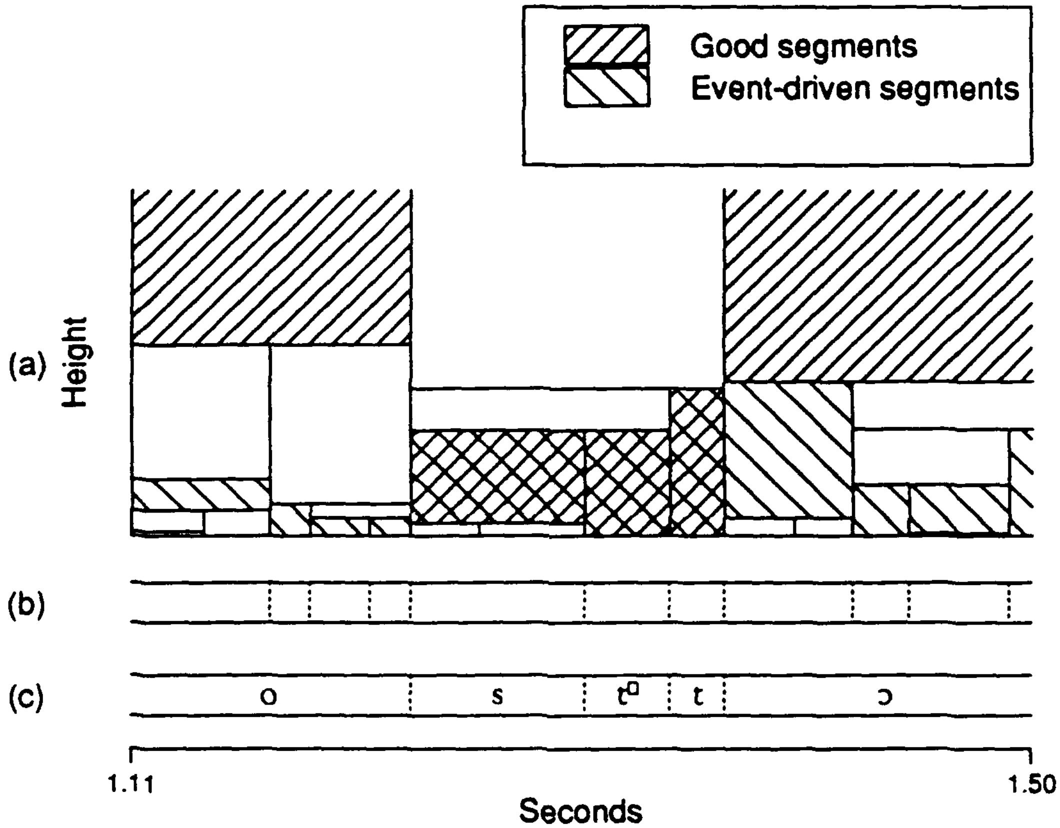


Figure 3.2: Evaluating the segmenter. (a) Multi-level segmentation. Meanings of cross-hatching patterns as in legend. (b) Segmenter output. (c) Phonetic transcription. Displayed times measured from beginning of utterance. Good segments in MLS correspond with transcribed phonetic regions shown in (c). Segments in MLS determined by segmenter coincide with those in (b). For this case, there are five transcription labels, out of which the segmenter labels three as good. MLS height defined in text.

represent duration, we used both left and right child durations in the measurement set. For seed regions, which have no children, left and right child durations were arbitrarily set to 1/2 the total segment duration. We used left and right child durations instead of the total since they provide more detailed information about the segment and thus are likely to be more useful in discriminating among segment types. Note that the total segment duration is the sum of the child durations so no information about segment duration is lost by using the child durations.

Because the process of building the MLS continues until the full utterance is spanned by a single segment, the MLS includes segments whose duration is so long that they must be merge segments. We found that over 99.5% of good and split segments are shorter than 500 ms. Consequently, almost all those longer than 500 ms are merge segments. Thus, the segmenter labels all segments longer than 500 ms as merge segments instead of determining class probabilities with the vector quantizer. Since the vector quantizer is not used to classify such segments, they are not included in the set of segments used to train the quantizer. Our reason for excluding very long segments from the vector quantization process is that it is preferable to allow a very few of these segments to be mislabelled rather than to depend on the training process to learn that almost all very long segments are merge segments. Also, by using this tactic, cluster centers are not wasted on regions of the space that are clearly merge segments. Because the clusters occupy a more compact space, there should be less vector quantization errc and the quantizer should be able to make finer distinctions among segments. For the same reasons, we also labelled segments whose heights were above a certain threshold as merge segments. Finally, since the segmenter does not compute the likelihood that seed regions are good segments as opposed to merged ones, seed regions were not included in the VQ training set.

Height and duration are not sufficient for classifying segments. This is clear from Figure 3.3, which is a copy of Figure 3.1 and illustrates the good segments

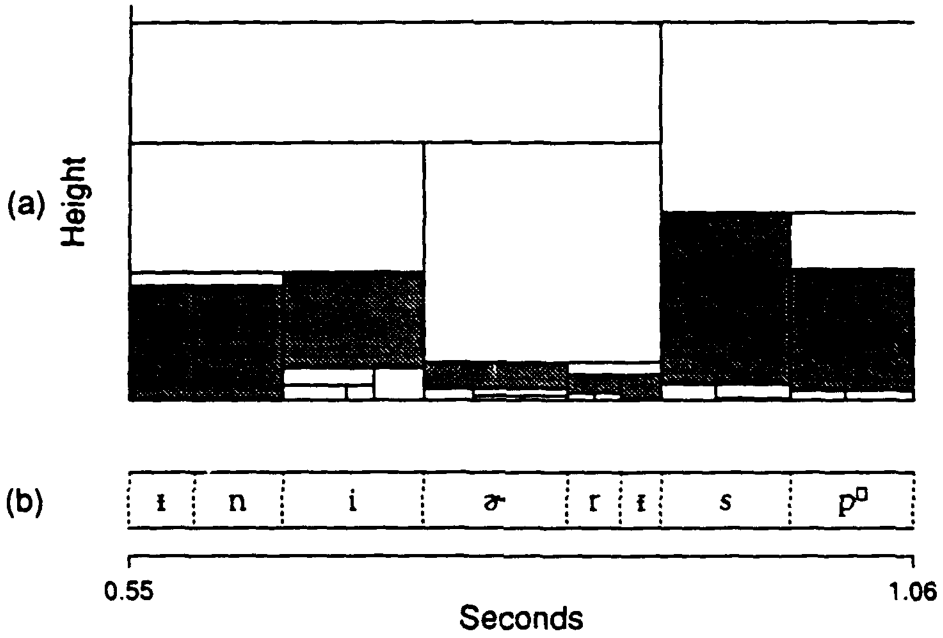


Figure 3.3: Heights and durations of good segments. (a) Multi-level segmentation. (b) Phonetic transcription. Displayed times measured from beginning of utterance. The shaded regions are "good" segments. Note that heights of good segments (denoted by the bottom horizontal segment edges) may be less than those of split segments (e.g., the segment associated with /n/ is lower than that of split segments associated with /i/). While no examples are shown here, good segment heights may exceed those of merge segments. Likewise, duration alone cannot be use to distinguish the segment types (e.g., the merge segment occupied by the /ɪ/ and /n/ labels is shorter than that occupied by the /i/ label.

in an MLS. For instance, the height of the good segment associated with the /ɪ/ and /n/ labels is less than that of split segments associated with the /i/ label. The problem is that thresholds of segment height and duration for discriminating segment types are very dependent on the acoustic qualities of the segment types. For instance, a segment consisting of a vowel and semivowel might be distinguished from a good segment consisting of only a vowel by its duration and by a slightly greater spectral change in the merge segment. Conversely, a merge segment consisting of a stop burst and vowel might have a similar duration to that of a good segment consisting of just a vowel but would likely have a much greater spectral change. Additionally, the difference spectra for the vowel-semivowel and stop burst-vowel spectral changes might have very different shapes. To capture this variability, a more detailed measurement set must be used. In our experiments to determine the best measurement set, we considered measurements including

1. the first seven hair-cell envelope principal components (HCEPC's) averaged over all frames in the segment's left child,

2. HCEPC averages over the right child, and

3. differences in right and left child HCEPC averages.

Finally, we experimented with another set of measurements, the *maximum spectral deviations* (MSD's), for capturing within-segment spectral change. The maximum spectral deviation for each HCEPC reflects the maximum amount that the principal component deviates from a straight line trajectory over the course of the segment. Let $h_{i\tau}$ be the ith HCEPC for frame τ of the segment, T the number of frames in the segment and q the number of HCEPC's used in the computation. The computation is:

1. For each $h_{i\tau}, 1 \leq i \leq q$ and $1 \leq \tau \leq T$, set

$$\hat{h}_{i\tau} = h_{i1} + (h_{iT} - h_{i1})\frac{\tau - 1}{T - 1}. \tag{3.7}$$

We will refer to this as the *linearly interpolated estimate* of $h_{i\tau}$ since if the trajectory of $h_{i\tau}$ between the first and last frames of the segment is a straight line, $\hat{h}_{i\tau} = h_{i\tau}$.

2. Determine the frame $\hat{\tau}$ whose HCEPC's deviate maximally from the straight trajectory according to the formula

$$\hat{\tau} = \arg\max_{\tau} \sum_{i=1}^{q} w_i (h_{i\tau} - \hat{h}_{i\tau})^2$$

where w_i are weights that account for the fact that the scale of $(h_{i\tau} - \hat{h}_{i\tau})^2$ is dependent on i, the index of the HCEPC. In particular, the scale increases with i because, as we discussed in Section 2.3.3, higher-index HCEPC's are related to finer spectral shape features and these tend to vary more over the course of the segment than do lower-index HCEPC's. Thus, the weights help ensure that deviations in the higher-index HCEPC's do *not dominate* in determining the maximum-deviation frame. Appendix A describes the algorithm used to determine the weights.

3. For all i, $1 \leq i \leq q$, compute MSD_i as

$$\text{MSD}_i = \hat{h}_{i\hat{\tau}} - h_{i\hat{\tau}}.$$

We also tested two related measurements. The maximum spectral deviation RMS (MSDRMS) is defined as

$$\text{MSDRMS} = \sqrt{\sum_{i=1}^{7} \text{MSD}_i{}^2}$$

and reflects the magnitude of the maximum spectral difference. The maximum spectral deviation spectrum (MSDS) is defined as

$$\text{MSDS}_i = h_{i\hat{\tau}}$$

and characterizes the spectrum on the frame with the maximum deviation.

We hypothesized that the MSD coefficients would be useful in distinguishing good segments for which there is relatively large but smooth spectral change (such as those associated with diphthongs) from merge segments for which there is large but abrupt spectral change (such as those associated with a stop burst-vowel sequence). In the first case, we hypothesized that the values of the coefficients would be relatively low because the HCE trajectories would be quite smooth and well-approximated by the linear model of Equation 3.7. In the second case, the abrupt change would presumably lead to large ·oefficient values.

Table 3.2 summarizes the results of our experiments with different measurement sets and codebook sizes. As stated above, the goal of these experiments is to find the measurement set and codebook size that produces the largest number of good segments from a given set of utterances. The results were obtained for the segmenter test set of ten male TIMIT speakers. In all cases, the left and right child spectral measurements are based on the first seven HCEPC's. These account for over 90% of the variance of the hair-cell envelopes and we decided arbitrarily not to use a higher number in our experiments. For the other measurement types, only the number of measurements that produced the highest number of good segments is included in the table, since this is the criterion used to evaluate segmenter performance. For example, the best results with the MSDS coefficients were obtained when four were used.

The results are difficult to interpret in some cases because of interactions among factors that are hard to explain. For example, sets (2) and (3) differ in the fact that the former includes the left child principal components (PCL's) and principal component differences (PCD's) while the latter replaces the PCD's with right child principal components, thus encoding the same information in a different way. The relative number of good segments in the two schemes seems to be dependent upon K, the size of the codebook. A similar phenomenon is seen in comparing sets (4) and (5), which differ in the same manner. In general, the results seem to be quite sensitive to the codebook size.

	K	Good	Merge	Split	Seg/L	G/L
(1) PCL$_1$-PCL$_7$, PCD$_1$-PCD$_7$, DurL, DurR, H	128	911	366	618	1.01	.49
	256	921	367	603	1.01	.49
(2) PCL$_1$-PCL$_7$, PCD$_1$-PCD$_7$, DurL, DurR, H, MSDRMS	128	904	388	534	.98	.49
	256	894	394	489	.95	.48
(3) PCL$_1$-PCL$_7$, PCR$_1$-PCR$_7$, DurL, DurR, H, MSDRMS	128	884	401	491	.95	.47
	256	923	381	527	.98	.49
(4) PCL$_1$-PCL$_7$, PCD$_1$-PCD$_7$, DurL, DurR, H, MSD$_1$-MSD$_3$	128	890	390	515	.96	.48
	256	924	377	550	.99	.49
(5) PCL$_1$-PCL$_7$, PCR$_1$-PCR$_7$, DurL, DurR, H, MSD$_1$-MSD$_3$	128	918	391	478	.96	.49
	256	915	381	557	.99	.49
(6) PCL$_1$-PCL$_7$, PCD$_1$-PCD$_7$, DurL, DurR, H, MSDS$_1$-MSDS$_4$	128	905	382	537	.98	.48
	256	**947**	**374**	**564**	**1.00**	**.51**
	512	919	384	481	.95	.49

Table 3.2: Effect of codebook size and measurement set on segmentation. Results are for the segmenter test set, consisting of utterances from ten male TIMIT speakers. K denotes number of clusters in codebook, Seg/L denotes number of segments per label. G/L denotes fraction of labels that are segmented into good segments. Measurements: PCL = left child hair-cell envelope principal component averages, PCR = right child hair-cell envelope principal component averages, PCD = differences between PCR and PCL, DurL = left child duration, DurR = right child duration, H = height of segment in MLS, MSDRMS = maximum spectral deviation RMS, MSD = maximum spectral deviation, MSDS = maximum spectral deviation spectrum. There were 1870 transcription labels in data set. The merge/good likelihood threshold Λ was set to 1.0. The row in boldface corresponds to the configuration chosen for use in the segmentcr.

Since we do not have a good explanation for this, these fluctuations might be attributable to chance.

There also appears to be a positive correlation between the number of good segments and the total number of segments. Thus, it is possible that if Λ were adjusted so that each measurement set produced an equal total number of segments, the number of good segments produced would show less variation.

From the table, it can be seen that the highest number of good segments, 947, was obtained with a codebook size of 256 and a measurement set including left and right child durations, segment height, the first seven left child HCEPC averages, the first seven HCEPC average differences between left and right children, and four MSDS coefficients. Thus, we used this measurement set in the segmenter. The ratio of good segments to transcription labels for this choice is .51. Thus, for the segmenter training set about half of the transcription labels are associated with good segments.

As predicted, a value of $\Lambda = 1$ led to roughly one segment produced per transcription label, indicating that the algorithm was not biased towards either merged or split segments with this choice. For building subword models we used a value of $\Lambda = .33$ which led to 1.4 segments produced per label. We discuss the reasons for this choice of Λ in Section 3.5, where we describe phone modelling within the segment-based HMM framework.

3.5 Subword Modelling

In this section, we describe the training of subword segment-based HMM's. These models were used in the phonetic recognition and word spotting experiments described in Chapters 4 and 5, respectively. We specifically refer to these as subword models rather than phone models because, as we discuss below, models for phone pairs, which we will call *biphones*, were built as well. The training set used to build these models consisted of 2150 utterances from each of the 430 TIMIT male speakers and a total of 473 utterances from ten male speakers in the VOYAGER corpus.

The fact that the segmenter produces both split and merged segments as well as good ones must be taken into account when training subword models. We illustrate the problem in Figure 3.4, which depicts the segmenter's output on part of an utterance. The arrows indicate the associations made between the segments and lexical labels. As discussed in Section 1.3, the segments associated with each label are used to train the label's HMM. A segment is associated with a label if more than 1/2 the segment overlaps the label or if more than 1/2 the label overlaps the segment. This rule is somewhat arbitrary and its possible effect on system performance will be discussed further in Section 3.8. One of its qualities, however, is that each segment is associated with at least one label and vice-versa. Thus each segment and each label are used in the training process.

We should note here that phonetic transcription labels will usually be represented using the International Phonetic Alphabet (IPA) and this representation will be used in displaying transcriptions (as in Figure 3.4). In the text, we will distinguish transcription labels from acoustic models for them by denoting the latter with boldface ARPABET labels. However, this convention will be overridden in some of the figures presented for the purpose of making clearer illustrations. Table 3.3 displays the equivalences between the two sets of labels.

Four types of associations are shown in the Figure 3.4. They are listed in Table 3.4. The table's "Arrow configuration" column schematizes the configuration of arrows in the figure corresponding to each type of association. The "Examples" column indicates the phone label(s) in Figure 3.4 that is pointed to by the arrows in question. Finally, the "Segment type" column refers to whether the segment(s) is actually a merge, good, or split segment, although this information is not used directly in the training process.

In cases where one or more segments are associated with a single phone, the sequence of segments so associated is used to train the model for that phone.

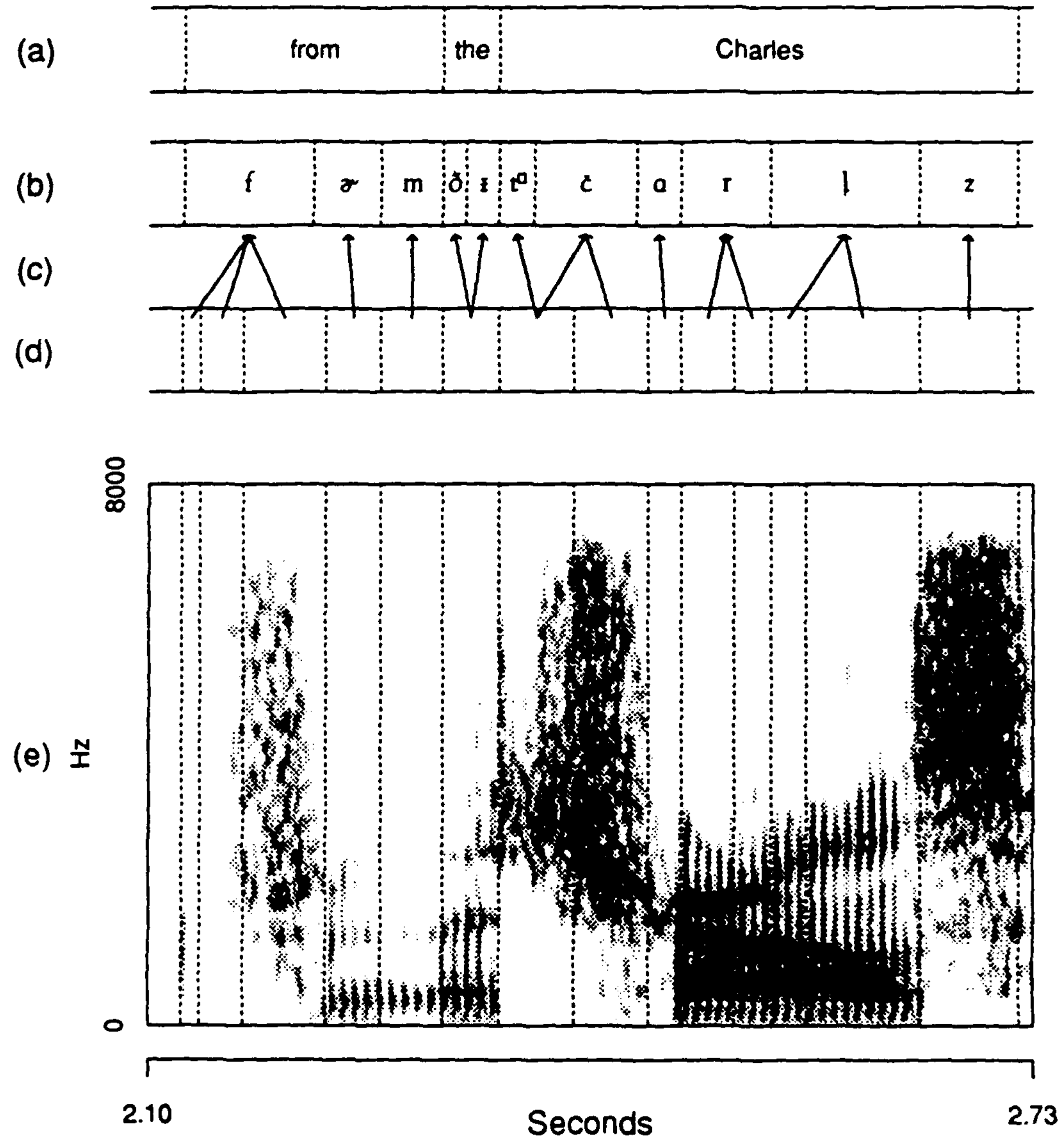


Figure 3.4: Segmentation of an utterance. (a) Orthographic transcription. (b) Phonetic transcription. (c) Associations between segmentation and phonetic transcription, described. (d) Segmenter output. (e) Spectrogram (overlaid with segment boundaries). Displayed times measured from beginning of utterance.

i	i	beat	s	s	see	w	w	wet
ɪ	ih	bit	š	sh	she	r	r	red
e	ey	bait	f	f	fee	l	l	let
ɛ	eh	bet	θ	th	thief	y	y	yet
æ	ae	bat	z	z	zoo	m	m	meet
ɑ	aa	Bob	ž	zh	Gigi	n	n	neat
ɔ	ao	bought	v	v	vice	ŋ	ng	sing
ʌ	ah	but	ð	dh	thee	č	ch	church
o	ow	boat	p	p	pea	ǰ	jh	judge
ʊ	uh	book	t	t	tea	h	hh	heat
u	uw	boot	k	k	key	l̩	el	bottle
ɝ	er	Burt	b	b	bee	m̩	em	bottom
ɑ^y	ay	bite	d	d	Dee	n̩	en	button
ɔ^y	oy	Boyd	g	g	geese	ɾ̃	nx	inner
ɑ^w	aw	bout	p°	pcl	(*)	ʔ	q	bottle
ə	ax	about	t°	tcl	(*)		epi	(†)
ɨ	ix	rabbit	k°	kcl	(*)		pau	(‡)
ɚ	axr	manner	b°	bcl	(*)			
ü	ux	avenue	d°	dcl	(*)			
			g°	gcl	(*)			
			ɾ	dx	butter			

Table 3.3: IPA and ARPABET phones. First column is IPA, second is ARPABET, third is a sample word. The stop symbols p, t, k, b, d and g are used to transcribe stop burst and aspiration while the closure symbols p°, t°, k°, b°, d°, g° are used to transcribe the stop closures. Special symbols: (*) closures, (†) epenthetic silence, (‡) interword silence.

Association	Arrow configuration	Examples	Segment type(s)
one-to-one	↑	ɚ, m, ɑ, z	good
many-to-one	↗↖	f, r, l̩	split
one-to-many	↖↗	ðɪ	merge
many-to-many	↖↗↖	t°č	merge-split

Table 3.4: Types of segment/label associations. "Arrow configuration" column schematizes the configuration of arrows in the Figure 3.4 for the corresponding association type. "Examples" column indicates the phone label(s) in Figure 3.4 that is pointed to by the arrows in question.

		Segment sequence length				
		1	2	3	4+	Total
Label	Phone	53.9	23.9	7.0	2.6	87.4
sequence	Biphone	6.6	3.9	1.1	0.4	12.0
type	Triphone	0.3	0.2	0.1	0.0	0.6
	Total	60.8	28.0	8.2	2.9	100

Table 3.5: Segment/label associations tabulated over transcription labels, expressed as percentages. Computed from 89946 transcription labels in subword model training set.

This is no different from a frame-based HMM, for which all associations are many-to-one because each frame is shorter than the shortest phonetic region. However, the existence of sequences of one or more segments associated with a sequence of more than one phone label must be modelled as well. We deal with this problem by augmenting our subword label inventory with phone label sequences that are often involved in these one-to-many and many-to-many associations. Thus a model for the biphone /ðɪ/ is included in our inventory, for example. A similar strategy was used in [LeMaire 89] to account for phones often merged in their digit recognition task. According to our convention, we will refer to this model's label as **dh-ix**. In general, biphone labels will be hyphenated.

The need to include such models is a potential drawback of a segment-based HMM approach because the typical frequency of occurrence of a sequence of several labels in the training set tends to be much smaller than that of a single label. Thus, there may not be enough data associated with some label sequences for making good model parameter estimates. It is for this reason that we biased the segmenter away from merge segments and towards split segments. As we mentioned in Section 3.4.4, the bias led to 1.4 segments being produced per transcription label.

Table 3.5 illustrates the extent to which segmenter variability must be accounted for in building a model inventory. It tabulates the percentages of labels involved in segment/label associations of various types as observed in the subword model training set. Thus, for example, each biphone associated with a single segment would count as two labels involved in a two-label/one-segment association. As can be seen from the table, 12.0% of transcription labels should be modelled as biphones and 0.6% as triphones. Thus, a total of 12.6% of labels are involved in one-to-many or many-to-many associations. The remainder of sequences are involved in either one-to-one (53.9%) or many-to-one (33.5%) associations.

Note that the subword model training set, upon which these statistics were compiled, includes utterances from the forty TIMIT speakers used to train the segmenter and the ten TIMIT speakers used to set segmenter parameters. Since the segmenter was trained to produce one-to-one associations, the results might overestimate the probability of this type of association on new data. However, as the utterances used to train and test the segmenter comprise less than 10% of the total subword model training set, this effect should be small.

Because of the small number of labels associated with triphones, we decided to add only biphone models to the phone inventory. We will refer to instances of biphones that are involved in one-to-many or many-to-many associations as merged biphones.

We did not build models for each biphone that could potentially be merged because the required number of models would have been very large, leading to great complexity in the phonetic recognition and word spotting systems in which the models were used. In fact, as shown in Table 3.6, out of 57 × 56 = 3192 potential biphones, 2180 were observed at least once in the training set. Out of these, 803 were merged at least once. Thus, even if we devoted models only to biphones that were merged at least once, using the reasoning that these would be the most likely to occur in the test data, computation would have been prohibitive.

To be included in the model inventory, a biphone had to meet two criteria. First of all, the number of tokens of the biphone in the training set had to

Potential biphones	Observed biphones	Observed merged biphones
3192	2180	803

Table 3.6: Biphone merge statistics. Observed biphones refers to the number of distinct biphone labels observed in subword model training set. Observed merged biphones refers to number of these merged at least once.

exceed an arbitrary threshold of 25 so that there would be enough training data for the model. Note that we used the *total* number of biphone tokens as the threshold, not the number of merged tokens. As we discuss in Section 3.7, all tokens of a given biphone were included in the training set for that biphone's model, not only those tokens that were merged.

Since each biphone model is used to account for instances where a biphone is merged, the second criterion for inclusion was the estimated frequency of such an occurrence. Because relative biphone frequencies are probably vocabulary-dependent and the test data were drawn from the VOYAGER corpus, biphone frequencies were collected from this corpus in making the estimate. In particular, 780 VOYAGER utterances from a set of sixteen female speakers and 149 utterances from a set of three male VOYAGER speakers were used to make estimates of relative biphone frequencies. The male speakers used were a portion of the subword model training set and thus were disjoint from speakers used in the test set. We assumed that there would be no great difference between male and female vocabularies so that this set would provide good estimates for the test speakers' biphone frequencies. These data will be referred to as the *language* training set. For each biphone a, $\widehat{\text{Merge}}(a)$, its estimated merged biphone frequency is given by

$$\widehat{\text{Merge}}(a) = N_{\mathcal{L}}(a)\frac{\text{Merge}_{\mathcal{S}}(a)}{N_{\mathcal{S}}(a)} \tag{3.8}$$

where $N_{\mathcal{L}}(a)$ is the number of occurrences of a in the language training set, $\text{Merge}_{\mathcal{S}}(a)$ is the number of merged occurrences of a in the subword model training set, and $N_S(a)$ is the total number of occurrences of a in the subword model training set. Thus, the biphone's relative frequency is estimated from the language training set and the tendency of it to merge is estimated from the subword model training set and the two are multiplied to obtain the estimate of the total number of occurrences. All biphones whose estimates were above an arbitrary threshold and which occurred often enough in the subword model set were included in the model inventory.

In all, 84 diphone models were included along with the 57 phone models for a total of 141 models. We estimated that these would model all but about 2% of the segment sequences observed in the test set. Table 3.7 summarizes the numbers of each type of model and Tables 3.8 and 3.9 enumerate the labels of all the models used.

Phone models	Biphone models	Total models
57	84	141

Table 3.7: Model inventory statistics.

In principle, the remaining sequences could have been modelled by labels of the form $\star$-x or x-$\star$ where x represents some phone label and $\star$ represents a "wildcard" phone label. The first of these models would be trained by all biphones whose right label is x and the second would be trained by all biphones whose left label is x. In a recognizer or word spotter, these wildcard models

ao	ax	eh	r	em	b	k	z	ch	kcl
ow	uh	ey	er	m	p	q	s	bcl	pau
aw	ux	ae	w	en	d	v	zh	pcl	epi
aa	uw	iy	y	n	dx	f	sh	dcl	
ay	ix	axr	el	ng	t	dh	hh	tcl	
ah	ih	oy	l	nx	g	th	jh	gcl	

Table 3.8: Phone model inventory.

aa-axr	b-axr	dh-ax	ix-n	n-kcl	q-ih	t-r
aa-n	b-ey	dh-ix	ix-nx	n-m	q-ix	t-s
aa-r	b-r	dx-ix	ix-q	n-q	q-iy	tcl-b
ae-ng	bcl-b	dx-iy	iy-ix	n-tcl	r-aa	tcl-s
ao-l	d-ax	eh-axr	k-s	ng-kcl	r-ah	tcl-t
ax-l	d-axr	eh-n	k-w	nx-ix	r-ax	v-dh
ax-n	d-ix	eh-r	kcl-k	pcl-p	r-eh	w-ah
axr-ix	d-iy	epi-n	l-ow	q-aa	r-ix	w-ax
axr-r	d-s	er-ix	m-dh	q-ae	r-iy	w-eh
ay-ix	dcl-d	gcl-g	n-d	q-ah	s-tcl	y-ux
ay-tcl	dcl-dh	ih-axr	n-dcl	q-ay	t-hh	z-dh
b-ae	dcl-jh	ix-dx	n-dh	q-eh	t-ix	z-epi

Table 3.9: Biphone model inventory.

could be used to build models of words that include biphones that do not have their own models. There would still be some ambiguity in determining a word sequence from a subword model sequence. For instance, if a **⋆-r** model is used to model both /ar/ and /or/, the words "cord" and "card" would be indistinguishable. However, higher order semantic and syntactic information could be used to deal with the ambiguity. While we would have utilized such a strategy if we had applied the segment-based HMM to a large-vocabulary problem, for simplicity we have chosen to ignore the problem of unmodelled merged biphones in our work.

Figure 3.5 charts the tendencies of particular biphones in the model inventory to be merged into a single segment. The tendencies are illustrated for the thirty biphones most likely to be merged. For each biphone, the tendency is expressed as the fraction of occurrences of the biphone in the subword model training set that are associated with a single segment.

The labels most likely to be merged are often those that involve voiced stop releases (e.g., **b-axr**), those for which it is difficult to find a boundary between aspiration and frication (e.g., **t-s** and **k-s**), vowel-semivowel combinations (e.g., **r-ax**) and other pairs of phones which share properties so that

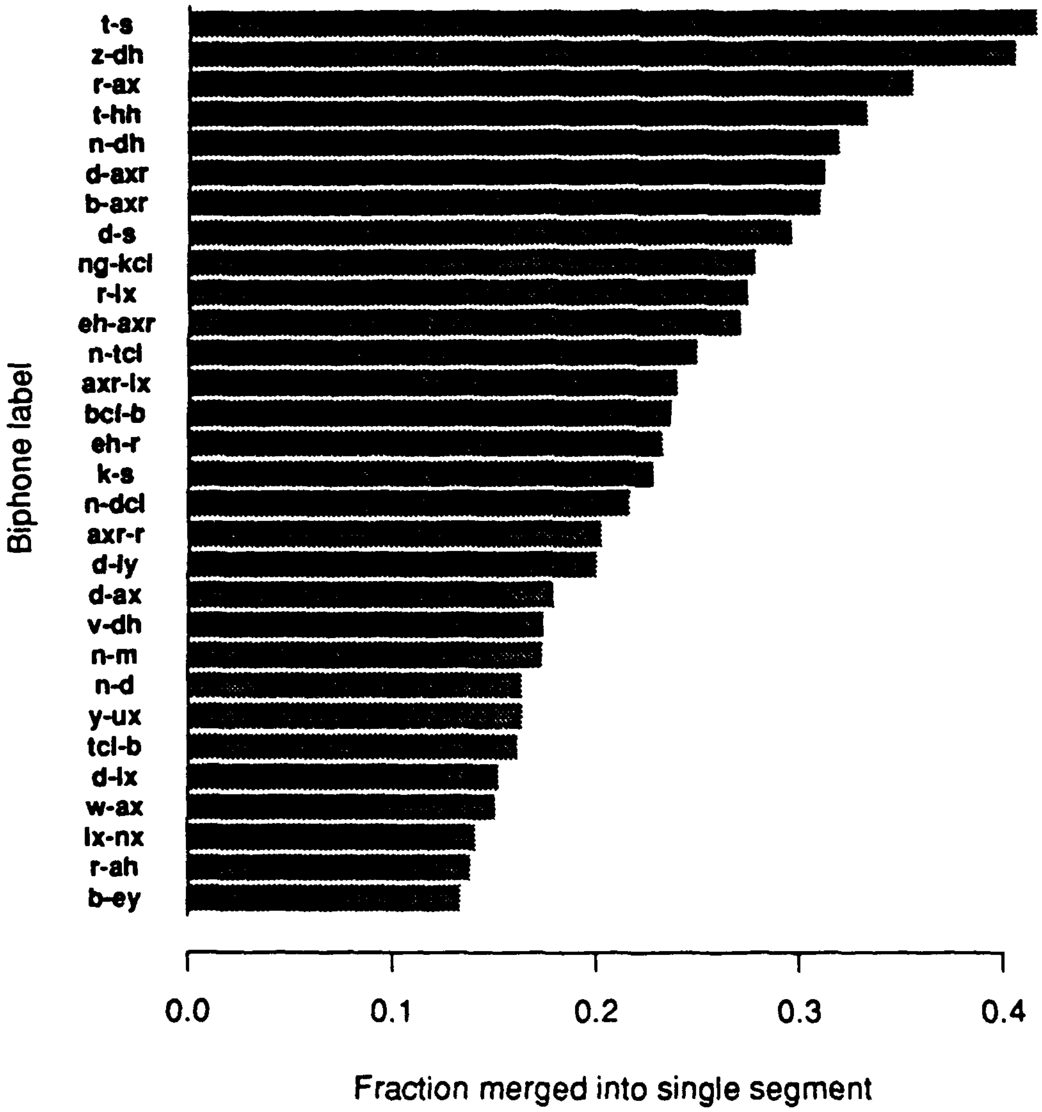


Figure 3.5: Biphones frequently merged into a single segment. Barchart includes thirty labels in biphone inventory that have highest tendency of being merged into a single segment. For each biphone, the tendency is expressed as the fraction of occurrences of the biphone in the subword model training set that are associated with a single segment.

finding a boundary between them is difficult (e.g., **z-dh**, **axr-r**, and **n-m**). Figure 3.6 displays examples of the three biphones that have the highest tendency of being merged into a single segment. From left to right, the biphones are /ts/, /zð/ and /rɔ/. We have used the IPA representations to identify the biphones since they are used in the phonetic transcriptions shown in the figure. The corresponding model labels are **t-s**,**z-dh** and **r-ax**. Note that the rightmost figure also includes the merged biphone /mð/.

In each of the displays, inspection of the spectrogram reveals acoustic cues to the presence of a phonetic boundary. In the leftmost display, there appears to be a double release for the /t/ that manifests itself as energy at low frequencies. At the boundary between the /t/ and /s/ transcription labels, this low-frequency energy is reduced while there is an onset of energy above approximately 7000 Hz. In the middle figure, the boundary between the /z/ and the /ð/ may be associated with a reduction of energy above 7000 Hz and the onset of energy below 2000 Hz. Finally, in the rightmost figure, the boundary between the /r/ and the /ɔ/ is probably associated with the onset of nasalization in the /ɔ/ which manifests itself as a broadening of F_1 and perhaps in the discontinuity in F_4. These hypotheses are based on previous spectrogram reading experience.

Thus, for these cases at least, the segmenter was not sensitive enough to detect these boundaries. We have used only this segmenter in the phonetic recognition and word spotting tasks discussed in Chapters 4 and 5. Thus, we have no basis for estimating the effect a segmenter that deletes fewer boundaries would have on performance in these tasks. Future work in this regard might be worthwhile.

Figure 3.7 charts the tendencies of particular biphones in the model inventory to be associated with multiple segments. The tendencies are illustrated for the thirty biphones most likely to be involved in many-to-many associations. As can be seen by comparing this figure to Figure 3.5, there is a large overlap between the biphones most likely to form one-to-many and many-to-many as-

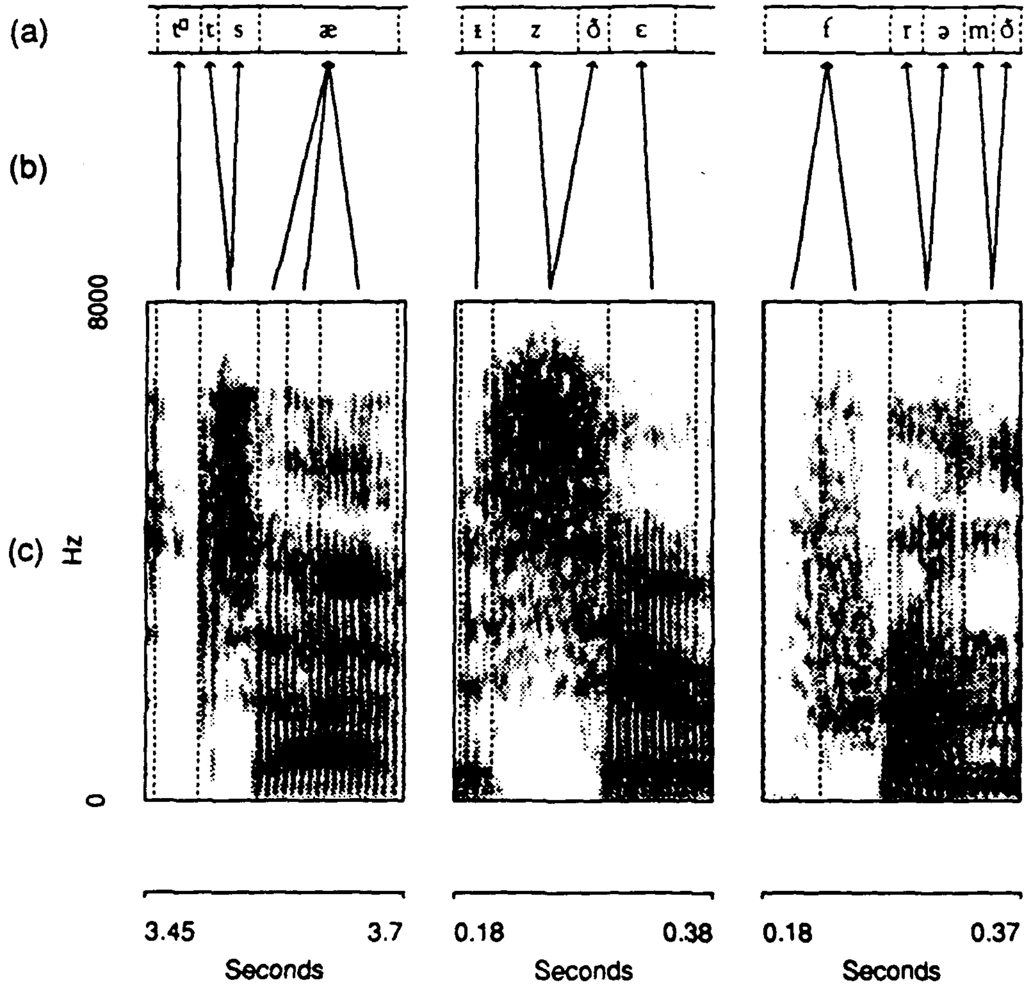


Figure 3.6: Examples of biphones merged into a single segment. (a) Phonetic transcription. (b) Associations between segments and phonetic regions. (c) Spectrogram with segment boundaries overlaid. Displayed times are measured from beginning of displayed utterance. The three examples from left to right show the merging of biphones /ts/, /zð/ and /rə/ whose corresponding model labels are **t-s**, **z-dh** and **r-ax**. Also note the merging of /mð/ in the rightmost figure.

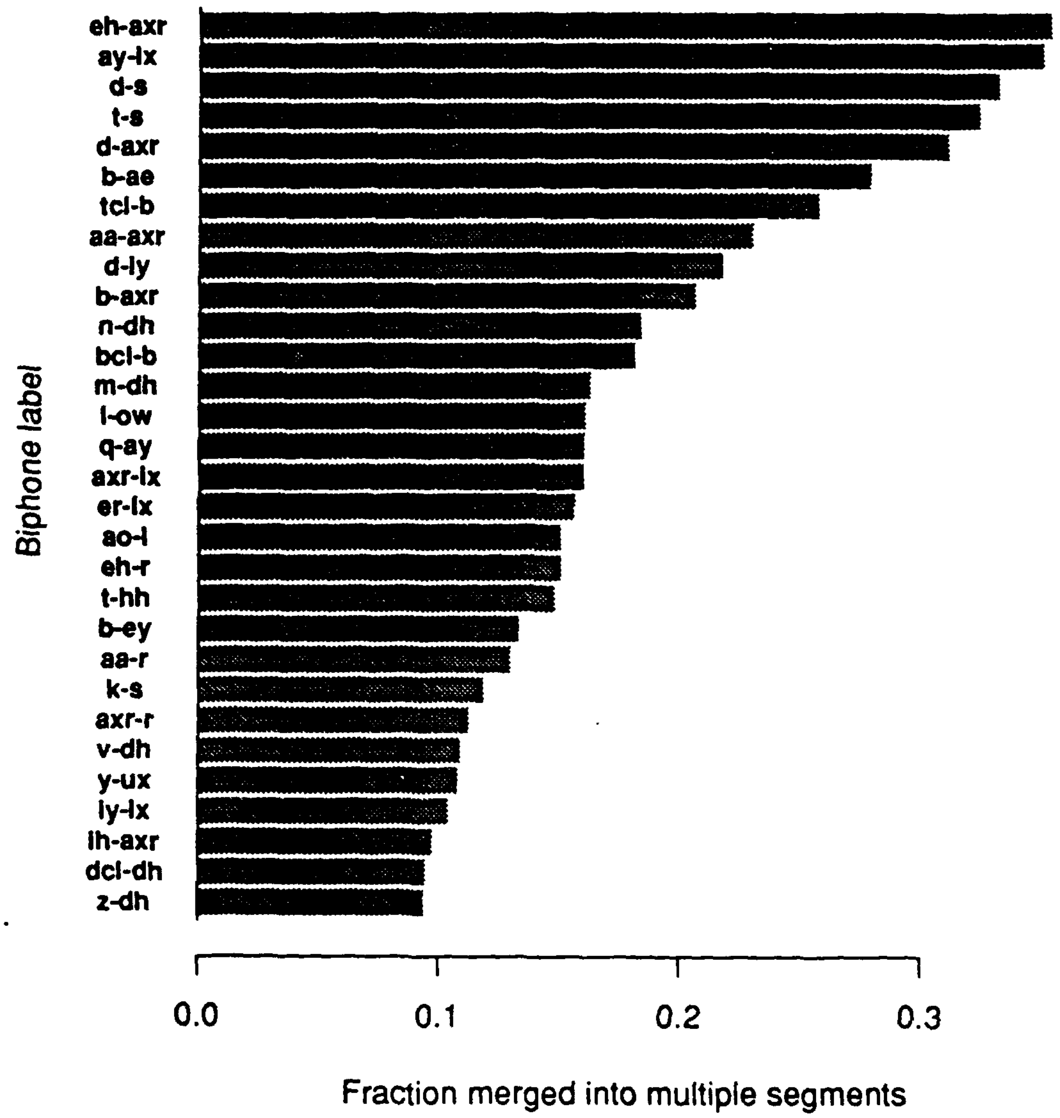


Figure 3.7: Biphones frequently associated with multiple segments. Barchart includes thirty labels in biphone inventory that have highest tendency of being associated with multiple segments. For each biphone, the tendency is expressed as the fraction of occurrences of the biphone in the subword model training set that is associated with multiple segments.

sociations. However, more vowel-semivowel pairs appear among the biphones with the highest tendencies of forming many-to-many associations. In general, the boundary between transcribed regions in such pairs is arbitrary because it is unclear where the acoustic properties associated with one phonetic label end and the next one begin. In fact, in some cases, transcribers used a convention of assigning 1/3 the syllable duration to the semivowel and 2/3 to a vowel [Zue 92]. Thus, it is not surprising that the transcription and segmenter boundaries do not line up in these cases. An example of this type is depicted for the biphone /ɛɚ/ in the leftmost display of Figure 3.8. The smoothness of the formants makes the determination of a boundary quite arbitrary. The segmenter finds a boundary at a point where high frequency energy rapidly decreases rather then at the transcription boundary.

The figure also displays examples of many-to-many associations for the biphones /ɑ^{y}ɪ/ and /ts/, which have high tendencies to form such associations. In the middle figure, a short segment is produced at the beginning of the /ɑ^{y}/'s phonetic region which captures the onset of voicing and another segment is produced that spans the remainder of this region and the /ɪ/ phonetic region as well. Thus, for this example, the many-to-many association might best be described as a sequence of a many-to-one association between the two segments and the /ɑ^{y}/ and a one-to-many association between the second segment and the biphone /ɑ^{y}ɪ/ The rightmost figure is similar in that a short region is found at the end of the /s/ phonetic region that captures the source change between the fricative and the vowel. The segment preceding it spans both the /t/ aspiration and the fricated portion of the /s/, similar to the one-to-many association for the biphone /ts/ shown in Figure 3.6. Thus, many-to-many associations can be classified into two types: those due to the arbitrary nature of the transcription boundary location for certain biphones and those that can best be characterized as a combination of one-to-many and many-to-one associations.

The need for biphone models distinguishes the segment-based HMM ap-

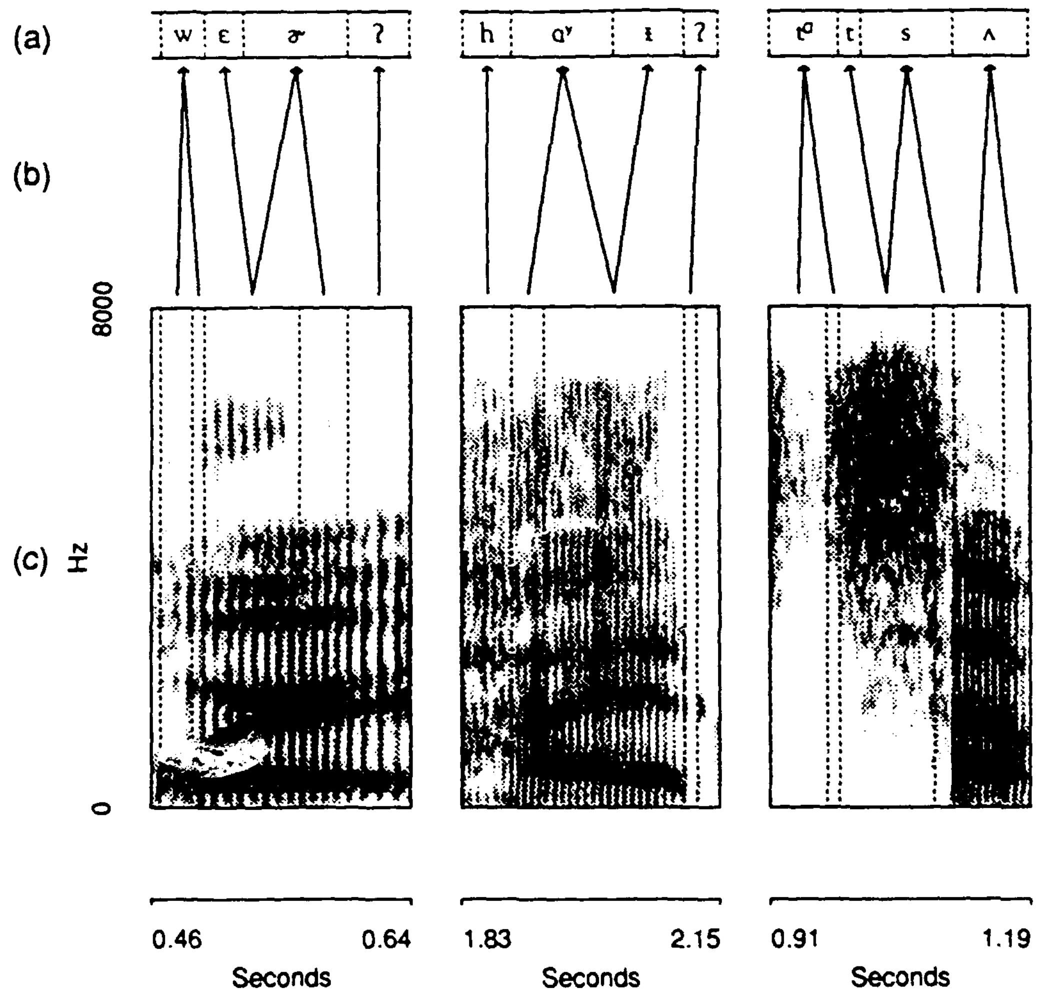


Figure 3.8: Examples of biphones associated with multiple segments. (a) Phonetic transcription. (b) Associations between segments and phonetic regions. (c) Spectrogram with segment boundaries overlaid. Displayed times are measured from beginning of displayed utterance. The three examples from left to right display biphones /ɛɚ/, /ɑ^{y}ɪ/ and /ts/ whose corresponding model labels are **eh-axr**, **ay-ix** and **t-s**.

proach from both the frame-based HMM and stochastic segment approaches to speech recognition. While the relative lack of training data for these models may pose a problem, they ma; be more appropriate units than phones to model in some cases. In particular, the fact that it may be difficult for the segmenter to find an acoustic boundary between two phonetic regions might indicate that it is more natural to treat the two regions as a single acoustic unit whose measurement PDF is a function of the identities of both phones. In this sense, biphone models are similar to the context-dependent phone models used in frame-based HMM's [Bahl 80, Schwartz 85, Chow 86, Lee 88].

3.6 Further Analysis of Segmenter Behavior

Table 3.10 reports the mean number of segments per sequence, mean duration per segment and other statistics for segments associated with phone labels of several different phonetic classes. The statistics were compiled for the subword model training set. Segments associated with biphones are included as well.

From the table, on? can see that the number of segments per phone produced by the segmenter is related to the amount of spectral change typical over the course of a phonetic segment for the particular phone. For example, voiceless stops, which include often include burst and aspiration portions, have more segments than voiced stops, which tend to be unaspirated. Likewise, unstressed vowels, which tend to have little spectral movement, have far fewer segments than diphthongs, with other stressed vowels falling in between.

The effect of phone duration may play a role in this, however. For instance, it might be more reasonable for the unreduced non-diphthongized vowels to have fewer segments per token than the voiceless stops, since the former tend to be relatively steady-state while the latter are often associated with two distinct acoustic events: a burst and an aspiration. However, there are fewer segments per stop token. Thus, the fact that the vowels are longer than the stops may account for the larger number of segments. Given that duration is one of

	Mean segments per sequence	Mean segment duration (ms)	Mean sequence duration (ms)	Number of sequences	Number of segments
Voiced stops	1.1	26	29	3796	4108
Reduced vowels	1.2	38	47	6477	7987
Voiceless stops	1.3	39	50	6436	8342
Affricates	1.3	53	70	1307	1733
Voiced fricatives	1.4	50	67	4865	6596
Nasals	1.5	41	63	6415	9843
Semivowels	1.5	45	68	8455	12742
Closures	1.5	46	68	12442	18450
Aspirants	1.6	44	68	965	1510
Multiphones	1.6	60	98	5590	9081
Voiceless fricatives	1.7	61	107	6516	11383
Glottal stops	1.8	35	64	1273	2352
Unreduced vowels	1.8	56	98	17745	31211
Diphthongs	2.4	62	149	1903	4559
Overall	1.5	49	76	84185	129897

Table 3.10: Segmenter statistics by phone class compiled for the subword training set. The multiphone category includes biphones and triphones. Voiced stops include both nasal and dental flaps. Unreduced vowels do not include diphthongs. The results are ordered by mean number of segments per sequence.

the measurements used in the segmentation algorithm, this would not be too surprising. On the other hand, the fact that duration *per segment* is dependent on phonetic type shows that the segmenter did not produce segments purely on the basis of total phone token duration.

One statistic not shown in the table is that /f/ and /θ/ have by far the highest average numbei of segments (2.5 for /f/ and 2.2 for /θ/) associated with them among the consonantal phones. This is because the spectrum is often uneven over the course of tokens of these phones, causing segment boundaries to be generated at points of spectral change. Figure 3.4 includes an example of this for /f/.

Finally, note that the average number of segments per sequence is 1.5. This is higher than the 1.4 segments per phonetic transcription label mentioned in the previous section, because sequences are associated with biphones and triphones as well as phones. Specifically,

$$\text{Segments per sequence} = \text{Segments per label} \times \text{Labels per sequence}.$$

This formula explains any apparent discrepancy between the two numbers.

Note that although, as originally stated, the goal of the segmenter was to produce ideal segmentation of one segment per transcription label, the segmenter tended to produce more segments for phones with greater spectral change and longer duration. Given that there not always a one-to-one relationship between well-defined acoustic events and phonetic transcription labels and that the measurements used to determine segments are based on spectral change and duration, this result is not surprising. We have dealt with the segmenter's variability by allowing the subword HMM's to have a structure flexible enough to model it. The HMM structure is discussed in detail in the next section.

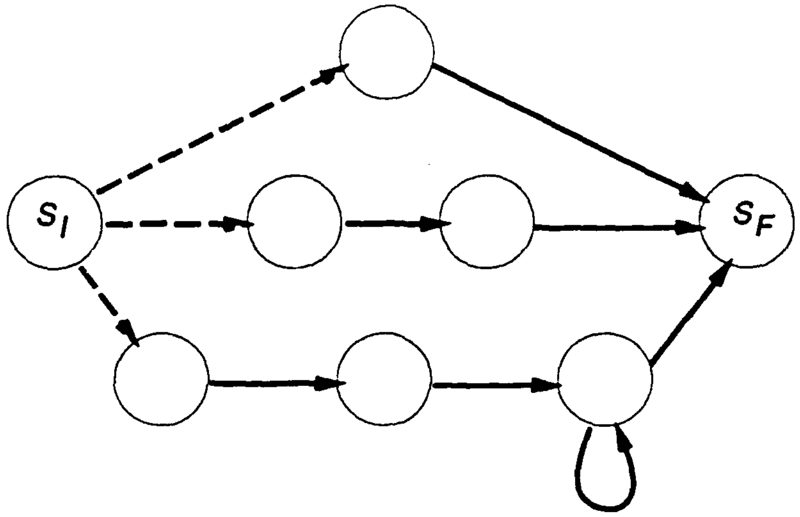


Figure 3.9: Phone/biphone model topology: S_I and S_F are initial and final states, respectively. Dashed and solid lines are null and segment-producing transitions, respectively.

3.7 Training the Subword Models

The general HMM topology for phone and biphone models is shown in Figure 3.9. In that figure, S_I and S_F are initial and final states, dashed lines represent null transitions, for which no segment is produced, and solid lines represent segment-producing transitions. The key characteristic of this type of model is that it has distinct branches, each one modelling tokens of a distinct number of segments. Let K be the number of branches in a model. Then for each k, $1 \leq k \leq K$, there is a branch with k states. Each branch k, $1 \leq k \leq K-1$, models tokens of k segments since it allows exactly k segment-producing transitions between initial and final states. For the remainder of the section, we will refer to a token of k segments as having length k.

The K^{th} branch models tokens of length K or greater since it includes a self-loop so that a token of any number of segments greater than or equal to K can be modelled by this branch. In general, for K odd, the self-loop is placed on state $(K+1)/2$ of the K^{th} branch, which is the middle state of the branch. For K even it is placed on state $K/2+1$ of the branch. We put the self-loop near the middle state of the branch because this state is trained by segments that occur near the middle of the token and we presume that this is the most acoustically stable region in the articulation of the phone or biphone. Thus, it might be more appropriate for a single state to model successive segments near the middle of a token through the use of a self-loop than it would be to model them near the edges of the token.

The number of branches in each model was determined by the distribution of the lengths of tokens used to train the model. Specifically, for each model, we computed a histogram of token lengths. For each token length L, we computed E_L, the number of tokens whose lengths are greater than or equal to L. Let N be the total number of tokens used to train the model. The largest L for which $E_L > 0.2N$ was used as the number of branches in the model, K. In other words, states in the K^{th} branch of each model were trained by at least 20% of the tokens in the training set associated with that model. In almost all cases, token length frequency falls with token length so that each branch is trained by at least as many tokens as used for the K^{th} branch. However, for some models, particularly biphone models, there were insufficient training data for some of the branches. We discuss solutions to this problem below. Table 3.11 tabulates counts of the number of branches used in the models. Table 3.12 lists each model and the number of branches it has.

Number of branches	1	2	3	4	Total
Count	18	100	21	2	141

Table 3.11: Numbers of branches ... phone/biphone models.

The parameters to be estimated in each model include the null transition probabilities, the self-loop probability and the state PDF parameters. While these could all in principle be estimated with the forward-backward [Jelinek 76]

ax	1	**hh**	2	**tcl-t**	2	**b-axr**	2	**ay-tcl**	2
b	1	**ih**	2	**r-iy**	2	**q-ih**	2	**dcl-jh**	2
d	1	**ix**	2	**aa-r**	2	**k-w**	2	**aa**	3
dx	1	**iy**	2	**r-eh**	2	**m-dh**	2	**ae**	3
epi	1	**jh**	2	**axr-r**	2	**w-ah**	2	**ao**	3
g	1	**k**	2	**axr-ix**	2	**t-r**	2	**aw**	3
nx	1	**kcl**	2	**s-tcl**	2	**t-s**	2	**ay**	3
dcl-d	1	**l**	2	**k-s**	2	**er-ix**	2	**en**	3
r-ix	1	**m**	2	**d-iy**	2	**t-hh**	2	**er**	3
r-ax	1	**n**	2	**kcl-k**	2	**z-epi**	2	**f**	3
dh-ix	1	**ng**	2	**y-ux**	2	**r-ah**	2	**ow**	3
d-ax	1	**p**	2	**gcl-g**	2	**n-dcl**	2	**q**	3
dh-ax	1	**pcl**	2	**n-d**	2	**v-dh**	2	**th**	3
d-ix	1	**r**	2	**w-ax**	2	**t-ix**	2	**uw**	3
q-ix	1	**s**	2	**d-s**	2	**eh-n**	2	**ux**	3
ix-nx	1	**sh**	2	**n-m**	2	**ax-l**	2	**aa-axr**	3
n-kcl	1	**t**	2	**pcl-p**	2	**dx-ix**	2	**tcl-b**	3
ix-dx	1	**tcl**	2	**b-ey**	2	**q-ae**	2	**ay-ix**	3
ah	2	**uh**	2	**ix-n**	2	**n-q**	2	**r-aa**	3
axr	2	**v**	2	**b-r**	2	**aa-n**	2	**l-ow**	3
bcl	2	**w**	2	**dcl-dh**	2	**epi-n**	2	**ae-ng**	3
ch	2	**y**	2	**tcl-s**	2	**d-axr**	2	**ao-l**	3
dcl	2	**z**	2	**eh-r**	2	**ax-n**	2	**b-ae**	3
dh	2	**zh**	2	**q-ay**	2	**iy-ix**	2	**oy**	4
eh	2	**bcl-b**	2	**ih-axr**	2	**dx-iy**	2	**pau**	4
el	2	**eh-axr**	2	**w-eh**	2	**q-ah**	2		
em	2	**n-tcl**	2	**q-eh**	2	**nx-ix**	2		
ey	2	**z-dh**	2	**ix-q**	2	**q-aa**	2		
gcl	2	**ng-kcl**	2	**n-dh**	2	**q-iy**	2		

Table 3.12: Number of branches in each model.

or segmental K-MEANS [Rabiner 89] algorithm, the particular structure of the models makes the training algorithm simpler. For $1 \leq k \leq K - 1$, the null transition probability for branch k is estimated as N_k/N where N_k is the number of tokens in the training set that have k segments and N is the total number of tokens. For branch K the formula is the same except that N_K is the number of tokens of at least K segments. The self-loop probability a_{ss} is estimated as $a_{ss} = C_{ss}/C_s$ where s is the index of the self-looping state, C_{ss} is the number of times in the training set there is a self-loop and C_s is the number of times the state with the self-loop is visited. To compute these counts, we note that for a token of $K + J$ segments, $J \geq 0$, each of the K segment-producing transitions between adjacent states is taken once, accounting for K segments. Thus, there are J self-looping transitions. There must be $J+1$ visits to state s, accounting for the J self-looping transitions and the one transition to the next state. Thus, if $\mathcal{K}$ is the set of all tokens in the training set that are of length K or greater and $K + J_i$ is the number of segments in token i,

$$a_{ss} = \frac{\sum_{i \in \mathcal{K}} J_i}{\sum_{i \in \mathcal{K}} (J_i + 1)}$$

Finally, to estimate the state PDF parameters, we use the fact that the position of each state in the model uniquely determines the segments used to train it. Specifically, each branch k, for $1 \leq k \leq K - 1$, is trained only from tokens of length k and the jth state in the branch is trained from the jth segments of these tokens. For example, in Figure 3.9 the right state of the second branch is trained by the second segment of each token of length two. For branch K, the self-loop makes the association between segments and states less straightforward but still deterministic.

In using this topology, we have made the assumption that segments from tokens of different lengths and different positions in the token are acoustically distinct. Thus each state should be trained from segments in a particular position from tokens of a particular length. For example, the segment in a one-segment token of /y/ is likely to be have greater duration and spectral change than either of the segments in a two-segment token of /y/. Additionally, the two segments in the two-segment token are likely to be different from each other acoustically since the first segment occurs at the start of the token and the second one at the end of it. These segments might be acoustically distinct, especially for a phone like /y/, over which there is a large spectral change.

There are a number of other topologies that could account for variable token lengths which do not make this assumption. We have not explored these alternatives.

We took special measures for training many of the biphone models since the number of merged biphone tokens available for training was judged to be insufficient. Thus, we added tokens to the training set that consisted of segments in the multi-level segmentation that were aligned with the biphone label but were not merged by the segmenter. Figure 3.10 illustrates two such tokens, which are represented by cross hatched segments in the multi-level segmentation.

These tokens were used to train the one-state branches of the **ix-n** and **r-ix** biphone models even though they were not produced by the segmenter. We refer to the tokens added to the training set in this way as *matched biphone tokens*. In general, for each utterance that included in its transcription the biphone to be augmented, the segment whose start and end points were closest to those of the biphone transcription boundaries was added to the training set. Thus, letting the start and end points of segment i be s_{ib} and s_{ie} and letting the start and end points of the biphone label be t_b and t_e, the segment was found for which $\mid t_b - s_{ib} \mid + \mid t_e - s_{ie} \mid$ was minimized and this segment was used to augment the training set for the biphone. In general, to augment the number of n-segment tokens so as to train the n-state branch of the model, a similar metric was used to find the sequence of n segments in the MLS that aligned best with the biphone label.

Because the matched biphone token segments are not produced by the segmenter, they may not be representative of the merged biphone segments

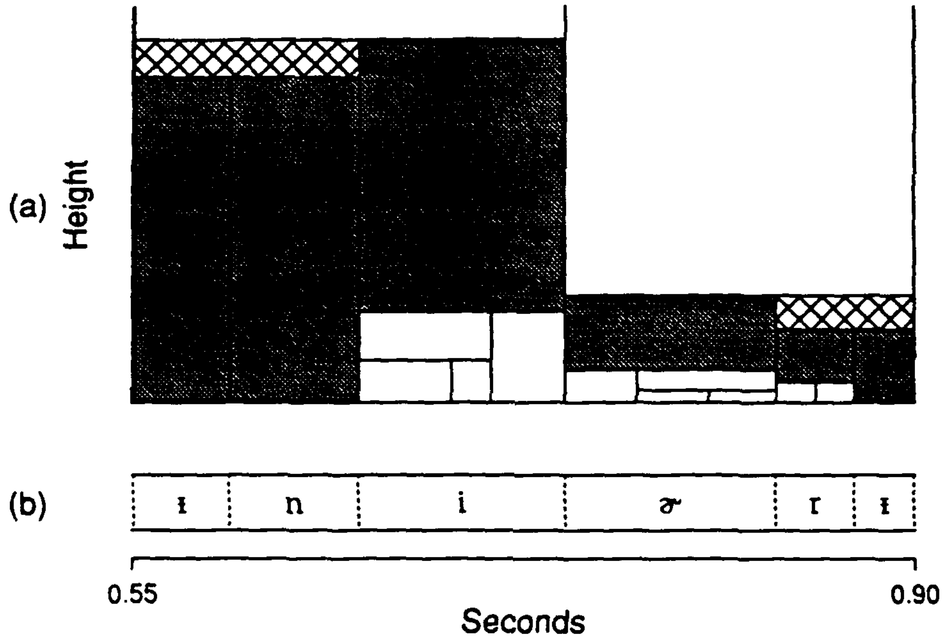


Figure 3.10: Augmenting the training set for a biphone model. (a) Training utterance multi-level segmentation. (b) Phonetic transcription. Shaded segments are produced by segmenter. Cross hatched segments are matched biphone tokens used to train models **ix-n** and **r-ix**.

being modelled. Thus, there may be a tradeoff between the benefit of increased training gained by adding these segments to the training set and the potential disadvantage of adding unrepresentative training data. We dealt with this issue by augmenting the training set by just the number of tokens needed to bring the total number of tokens used for training a branch of a biphone model to at most 25, an arbitrarily chosen threshold. Thus, for example, if there were originally M tokens in the training set for training a particular branch, then if $M < 25$, at most $25 - M$ matched biphone tokens were added to the training set. The matched tokens added were those whose segment boundaries matched most closely to the biphone transcription boundaries according to the distance metric introduced in the last paragraph. In the future, it might be useful to conduct experiments to determine whether the addition of matched segments was beneficial, or whether even more of them should have been added, using a higher threshold than 25.

We also used matched tokens to augment the training set for the oy model. In particular, there were fewer than 25 instances of /o^y/ associated with exactly one segment. Thus, we used matched tokens of exactly one segment to augment the training set for that model.

3.8 Summary and Discussion

In this chapter, we have described the segment-based HMM, which consists of an algorithm for producing acoustic segments and a set of HMM's for acoustically modelling sequences of these segments.

We have shown that while the stated goal of the segmenter is to produce one segment per transcription label, for our implementation this occurs for only about half the transcription labels. Tokens of phones which tend to be long or have large spectral change within their transcribed boundaries tend to be associated with a larger number of segments. Conversely, certain pairs of phones tend to be merged into single segments because of the difficulty in finding an acoustic boundary between phonetic regions.

Because the segmenter is not an end in itself but is a component of a speech recognition system, it is difficult to assess it without using it in a recognition task, which we do in the next two chapters. Also, we are unable to assess the effect that the segmenter's deviation from its intended behavior has on recognition results because we use only this segmenter in our work. However, we do find it encouraging that the deviations from intended behavior tend to be systematic and can usually be explained using acoustic-phonetic principles. Systematic behavior is easier to model than random behavior.

To improve the segment-based HMM, it would be worthwhile to optimize certain parameter settings and to investigate the validity of several of our assumptions. For instance, the number of segments produced per label was chosen arbitrarily to be 1.4 rather than 1.0 so as to reduce the number of merged biphones. Likewise, we made arbitrary choices in determining the inventory of subword labels, the minimum training data requirement for each model, the model topologies, and the rule for associating segments to labels. While we have provided reasons for these choices, we have not tested the effect of making different ones. For instance, it might yield better results to train the subword models automatically within the HMM framework as described in [Jelinek 76] instead of using our arbitrary association rule, which relies on the phonetic transcription for associating segments with labels.

More fundamentally, a segmenter could be built using an alternative model to the ideal segmenter described in Section 3.2. In particular, the justifications of using variable-length segments instead of frames, i.e., statistical independence among segments and a better framework for discriminating among labels, could be formalized in a mathematical model that could be used to determine a good strategy for segmentation and a good subword unit inventory. Ideally, such a model would include training set size as a parameter since for larger training sets, there would be less of a problem in training biphones and larger units so that it might be appropriate to produce longer and fewer segments corresponding to the larger units. The model would also have to account for the interaction of measurement set with discrimination power and segmentation. For example, a measurement set that does not take into account within-segment spectral dynamics might be appropriate for short segments or frames but not for larger segments within which there is greater spectral change. Such a model would avoid the ad hoc assumptions about the inventory of subword units made here. However, due to the complexity of the problem, building such a model might be difficult.

In our work, we have chosen to separate segmentation from recognition both because we believe that the tasks are conceptually different and so as to simplify the experimental paradigm. Another approach would be to integrate the segmenter more fully into the recognizer, as is done in the stochastic segment models outlined in [Ostendorf 89] and [Zue 89a]. The resulting segmentation could still be used in an HMM framework so that this approach would not be the same as adopting a stochastic segment model.

Finally, variable-length segments could be used within a recognizer that makes measurements on varying time scales. Measurements made on different scales might provide complementary information. For instance, measurements on both fixed-length frames and variable-length segments could be combined, the former useful for modelling events requiring fine temporal resolution and the latter useful for modelling more long-term phenomena such as formant trajectories over the course of a vowel or even pitch trajectories over the course of a syllable. Unlike the frame- or segment-based frameworks, this approach would entail measurements made over regions of the waveform that overlap in time. Thus, much attention would have to be paid to the problem of statistical dependence among measurements. However, if this problem were dealt with satisfactorily, the multi-scale framework could yield improvements in recognizer performance. Some proposals and initial work on this subject appear in [Digilakis 92].

In summary, there are many alternatives to those we have chosen for implementing a segment-based HMM. However, in any implementation, the designer would have to deal with the problems discussed in this chapter: determining a criterion for optimizing segmenter performance, developing a segmentation algorithm, and dealing with segmenter variability. In the next chapter we deal with another problem that would be faced in any implementation: that of determining the acoustic measurement set that characterizes each segment.

Chapter 4

Acoustic Modelling of Segments

This chapter has three major aims: to investigate how acoustic-phonetic knowledge can be represented in the set of measurements made on a segment, to compare the effectiveness of different measurement sets for phonetic recognition, and to establish the segment-based HMM as a feasible alternative to existing speech recognition approaches.

The choice of a measurement set for a variable-length acoustic segment is more complex than the choice for a fixed-length frame. In a frame-based system, the observation vector represents the spectrum, which is assumed to be stationary over the course of the frame, and typically includes measurements that represent spectral change in the frame's vicinity, e.g., [Gupta 87, Lee 88, Rabiner 89, Paul 91]. In a segment-based system, stationarity can no longer be assumed. Thus, spectral changes over the course of the segment must be modelled. Another important difference between frames and the acoustic segments we use is that, as we discussed in Chapter 3, segment boundaries may be good places around which to make measurements useful for phonetic discrimination while frame boundaries are generated periodically, independent of the signal. Thus, measurements made near these boundaries should be considered as well. Finally, there may be insufficient training data to provide good model parameter estimates if a large number of measurements is used.

Thus, techniques to reduce the dimensionality of the observation vector might be more important in a segment-based system than a frame-based one.

While determining the measurement set for a segment may be more complex, a segment-based system may also be a more convenient framework for representing acoustic-phonetic knowledge. In this chapter, we investigate how segmental measurements represent knowledge and how the representation can be improved. In particular, we

1. develop a regression model to predict formants from the MFSC's, evaluate the model fit with graphical techniques and show that non-linear transformations of the spectral coefficients can greatly improve the fit,

2. develop a method for clustering within-phone covariance matrices, show that the matrices cluster well by manner of articulation and use this fact to develop a refinement of multiple discriminant analysis that uses the clustering to produce manner-specific discriminants,

3. examine the relationship between the discriminant functions determined using this technique and distinctive features and show that for vowels there is a strong relationship between the two, and

4. show that measurements made just outside the segments associated with stop consonants contain information that allows discrimination of the stops.

To gauge the relative effectiveness of different measurement sets, we compare results on a phonetic recognition task. We show that performance can be improved substantially when measurements made just outside the one being modelled are added to the measurement set. Finally, we show that the phonetic recognition performance obtained by the segment-based HMM is in the range of that reported for other a,)proaches of similar computational complexity.

4.1 Previous Approaches

Measurement sets that have been used in segment-based systems can be classified into three types, which we will refer to as time-frequency, boundary-based spectral, and feature-based schemes. In some systems, hybrids of these are used.

Each segment in a time-frequency scheme is modelled as a sequence of a fixed number, M_A, of frames, where M_A may vary with the model's lexical label A. If each frame's spectrum is represented by p coefficients then the observation vector of each segment is of length pM_A. To account for variability in segment duration, a mapping is effected between frames of each segment in the test utterance and those of each segment model so that the number of frames represented in the test utterance segments and segment models are equal. The mapping method varies among different implementations. Examples of time-frequency schemes include systems described in [Bocchieri 86, Bush 87, Ostendorf 89, Digilakis 92]. The acoustic model in these systems is closest to that of frame-based HMM's in that most or all of the test utterance frames are used directly in computing the acoustic score. Thus, in both types of system, the signal is modelled as a sequence of short-term spectra, with no focus on segment boundaries.

In boundary-based schemes, spectral measurements are made at fixed locations relative to segment boundaries. The measurements might be frame-based or averaged over several frames. For example, [Leung 92] divides each segment into three equal-duration portions and represents each segment as the observation vectors averaged over each portion. In [LeMaire 89], the center frame and interframe differences at segment edges are used. In [Meisel 91], combinations of boundary-based measurements are used.

The distinction between the time-frequency and boundary-based schemes is in some sense arbitrary. It could be argued that boundary-based schemes are time-frequency schemes that use a simple mapping between test utterance and model frames, i.e., the frames are mapped based on their positions relative to segment boundaries. However, we make the distinction because the boundary-based scheme characterizes each segment with measurements made at particular points relative to segment boundaries rather then at nearly uniformly spaced points throughout the segment. One practical difference between the two approaches is that there tend to be fewer measurements made in boundary-based schemes. Making fewer measurements could be advantageous if the boundary-based measurements capture all the information necessary for discrimination since it alleviates the training problem. However, if the extra measurements made in the time-frequency scheme contain useful information, they may outperform boundary-based schemes.

Feature-based schemes use measurements that are often made over the full segment and tend to be more heterogeneous than those made in either of the other schemes. Examples of their use include the alphabet classification task described in [Cole 86] and the recognizer described in [Zue 89a]. In [Bush 87], the segment's peak low-frequency energy and duration and measurements of formants are included along with the time-frequency measurements so that this system uses a hybrid of time-frequency and feature-based schemes. Finally, our segmenter, described in Chapter 4, uses a feature-based scheme, including measurements such as the maximum spectral deviation over the course of the segment.

The distinction between boundary- and feature-based schemes is somewhat arbitrary in that the features of the latter scheme are often based on transformations of spectra measured at particular points in the segment. For instance, the features in [Zue 89a] include centers of gravity of hair-cell envelopes measured at particular instants. Thus, the two representations may contain the same information while using seemingly very different measurement sets. We make the distinction, however, because the two approaches represent different philosophies. In a feature-based scheme, the system designer incorporates his/her view of the set of useful measurements more directly than in the other two schemes, either by specifying the set of measurements used as in [Cole 86] or by specifying families of measurements and using automatic techniques to choose particular measurements from these families as in [Zue 89a].

For our work, we have chosen to rely on boundary-based measurements. These are simpler to implement than time-frequency measurements in that they do not require a function for mapping between test and model frames. Also, they should be able to take advantage of the fact that segment boundaries are supposed to be landmarks about which to make measurements useful for discrimination. We have chosen not to use a feature-based scheme mainly because the heterogeneity of measurements makes it more complex to develop such a scheme and to gain insight into system performance. Additionally, we were encouraged by the performance of the boundary-based measurements of [LeMaire 89] in a segment-based HMM to try a similar scheme. Finally, we show below that transformations of boundary-based measurements can be used to approximate formants and vowel distinctive features, underlining the fact that for some types of features the distinction between the two schemes is arbitrary.

4.2 The Baseline Measurement Set

Our first phonetic recognition and word spotting experiments were run with a measurement set similar to that used in [LeMaire 89]. We used this set to develop the software for phonetic recognition and word spotting and to produce an initial set of results for both tasks. Thus, we term this the baseline set.

In the following description of the set, b is the index in the utterance of the segment starting frame and e is that of the ending frame. The set includes fifteen measurements:

1. the first seven hair-cell envelope principal components (HCEPC's) measured at frame $\lfloor (b+e)/2 \rfloor$, the middle frame of the segment,

2. the first seven HCEPC differences measured at the right edge of the

segment between frames s_e and $s_e + 1$, i.e., the first seven components of the vector $\boldsymbol{h}_{e+1} - \boldsymbol{h}_e$, where $\boldsymbol{h}_i$ is the vector of HCEPC's measured at frame i, and

3. segment duration.

We used just the first seven HCEPC's since, as discussed in Chapter 3, they account for over 90% of variance. The right edge differences are used to measure spectral change at the segment edge, which are likely to be rich in discriminatory information, as discussed in Chapter 3. Finally, duration is included since it is known to be a valuable measurement for phonetic discrimination [Klatt 76].

4.3 Within-Segment Measurements Based on Mel-Frequency Spectra

For the remainder of this section, we describe measurements based on mel-frequency spectral coefficients (MFSC's) and their principal components (MF-SPC's). As discussed in Section 2.3, the MFSC's are based on a 25 ms Hamming window advanced 5 ms each frame. In the phonetic recognition experiments described in Section 4.8.3, we investigate the effect on performance of including different subsets of these measurements. We used the MFSPC rather then the HCEPC spectral representation for most of the phonetic recognition experiments because it is probably better suited to a Gaussian PDF model, as we discuss in Section 4.8.3.

To characterize within-segment dynamics, we measured the spectrum at three places within the segment:

1. The beginning of the segment was characterized by the segment's second frame MFSPC's. We will refer to this measurement set as the beginning MFSPC's (BMFSPC's). We used the second frame rather then the first so as to reduce the influence of the preceding segment since the purpose

of the BMFSPC's is to model the within-segment spectrum. We chose to use a single frame rather then averaging several frames, as in [Meisel 91] and [Leung 89], so as not to "smear" the spectral representation by averaging it over a long period. In any case, the Hamming window acts to average the spectrum so as to reduce the effects of signal non-stationarity on the measurements.

2. The middle of the segment was characterized by the segment's middle frame's MFSPC's (MMFSPC's).

3. The end of the segment was characterized by the segment's second-to-final frame MFSPC's. These will be referred to as End MFSPC's (EMFSPC's).

4. The MFSPC's averaged over the full segment were used as well. These will be referred to as average MFSPC's (AMFSPC's). We reasoned that, since they take the complete segment into account, these might characterize a segment better than measurements made at a specific location. Also, the relationship between the AMFSPC's and the other three subsets might provide information about the within-segment dynamics. For example, if the average spectrum resembled the spectra measured at the beginning and middle of the segment to a greater extent than that measured at the segment's end, it would suggest that the segment spectrum was relatively constant and then changed suddenly at the end.

All of these measurement sets used the first fifteen MFSPC's to represent the spectra. These account for over 98% of the variance in the training set's MFSC's. There is evidence that recognition performance tends to level off at around this number of spectral principal components [Brown 87] or MFSC's [Digilakis 92]. By way of comparison, Lee [Lee 88] used twelve and Digilakis [Digilakis 92] used eighteen mel-based cepstral coefficients in their work.

For each of the four within-segment spectra described above, we also considered the measurement $\log \sum_{i=1}^{40} \exp(\mathrm{MFSC}_i)$ where MFSC_i is the i^{th} MFSC. This was added to represent total energy and its form is based on the fact that the MFSC's are log energies. We will refer to this as the frame's energy. The begin, middle, end, and energies will be abbreviated as BENGY, MENGY, EENGY, and AENGY.

4.4 Formant Estimates Based on the Mel Spectrum

4.4.1 Motivation

As stated in Chapter 1, one of our key goals is to to use exploratory data analytic techniques to learn how acoustic-phonetic knowledge is represented in speech recognition systems and how the representation can be improved. This is the prime motivation behind the work presented in this section, in which we develop a multiple regression model to estimate formants from mel-frequency spectral coefficients.

The problem is of interest because formants play an important role in theories of speech production, perception and characterization. A sampling of the work done in perception and characterization includes studies of vowels [Peterson 52, Pols 73, Carlson 75, Klatt 82], semivowels [Espy-Wilson 87], fricatives [Soli 81], and stops [DeLattre 55]. Since it is believed that formants are important for phonetic discrimination, a good measurement set would presumably represent them. By building a regression model, we hoped to learn the form of this representation for MFSC's.

Specifically, if formants could be modelled well as linear combinations of MFSC's, there would be no need to include them in the recognizer's measurement set. If they were indeed important for discrimination, multiple discriminant analysis, which finds linear combinations of measurements that maximize discriminatory power, would presumably "discover" them. This would also hold true for a system that used full within-class covariance matrices to model the state PDF's. Decision boundaries in such a system can be arbitrary hyperplanes in the measurement space [Duda 73, pp 131-134]. The boundaries thus include hyperplanes that separate classes based on their formant measurements.

Conversely, if the linear model is poor, a system based on multivariate Gaussian models might benefit from the explicit use of formants in the measurement set. If phonetic discrimination is strongly related to differences in formant frequencies among different classes then adding the formants directly to the measurement would tend to make the decision boundaries more linear in the space used by the classifier. Thus, the classifier could have a more accurate model of the decision boundaries and could potentially perform better. We show below that the linear model is indeed poor.

Given this fact, a straightforward way of learning whether it is useful to add formant estimates to some other spectral representation would be to add the estimates of a formant tracker to the measurement set. Bush and Kopec [Bush 87] did exactly that and achieved no improvement in their digit recognition results when a formant tracker was added to a measurement set based on LPC spectra. The negative result was attributed to the fact that there was likely a strong correlation between the spectra and the formants. However, LPC analysis is designed to model vocal tract resonances [Atal 71] and we believed that the same results might not apply if the MFSC spectral representation was used.

Another reason there was no improvement might be that formant trackers are unreliable. Bush and Kopec point out that formants are "notoriously difficult to estimate reliably", and cite [Schafer 77] as a reference for this statement. In [Pols 73] and [Broad 89] this difficulty is pointed out as well. One problem is that formant trackers typically label formants explicitly. This could lead to catastrophic errors. For example, if F_2 is 1000 Hz but the spectral peak associated with it is missed by the tracker, then the peak associated with F_3

may be labelled as F_2. Also, formant trackers might exhibit unpredictable behavior where there is not much formant structure, such as in fricatives.

For these reasons, we did not use a formant tracker directly for adding formant information to the recognizer. Instead, we built a model to predict formants from MFSC's that involves non-linear transformations of the MFSC's and used this model to predict formant values for use in the recognizer. As we discuss below, the model has certain limitations. However, we believed it to be superior to a formant tracker for use in a speech recognizer because, by not explicitly labelling formants, it avoids the problems mentioned above.

Also, from an engineering point of view, it would be unwieldy and computationally expensive to add a separate set of features to the MFSC's for use in formant tracking. Thus, a model based on MFSC's is superior from this standpoint as well.

As we show in Section 4.8.3, addition of the formant estimates does not lead to a large improvement in phonetic performance. Nevertheless, the model building process is of interest in its own right. The poor performance of the linear model is an interesting result since it appears to contradict previous work that is similar to ours [Broad 89], as we discuss in Section 4.4.3. Also, we show that by including nonlinear transformations of MFSC's suggested by knowledge about the relationship between them and formant frequencies, the model can be greatly improved. This represents a novel approach to the problem.

4.4.2 Building the Model

We first built a multiple regression model to estimate F_2 from the 15 MFSPC's. We will describe in detail the process of building this model and then briefly present results obtained for a model of F_1, since the process was identical.

In general, a multiple regression model for predicting a response variable

y from a vector of regressors $(x_1, x_2, \ldots, x_p)$ is of the form

$$\hat{y} = a_0 + \sum_{j=1}^{p} a_i x_i \tag{4.1}$$

where $\hat{y}$ is the estimate of y given the regressors. The *regression coefficients* $a_0, a_1 \ldots a_p$ are determined from a training set of n vectors $(x_{i1}, x_{i2}, \ldots, x_{ip}, y_i)$, $1 \leq i \leq n$. It is assumed that the points are statistically independent of each other. The ordinary least-squares multiple regression method that w use determines the coefficients so that the mean squared error (MSE) $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ is minimized. See [Myers 88] for details on the procedure.

In our case, the regressors were the MFSPC's and the response variable was F_2. Our training set consisted of the five phonetically balanced ("sx") sentences from each of 23 male TIMIT speakers. We used a formant tracker provided with the ESPS signal processing package (released by Entropic Systems Inc., Washington, DC) to estimate formants F_k and bandwidths B_k, $1 \leq k \leq 4$. Formant estimates were produced with a frame advance of 10 ms. According to the ESPS reference manual, the tracker uses LPC analysis to estimate spectral resonance frequencies and enforces continuity constraints on formant frequencies with dynamic programming. We retained every tenth frame in the training set. Intervening frames were removed to reduce the statistical dependence among vectors in the training set that would have resulted from retaining sequences of adjacent frames.

It might seem contradictory to use formant tracker estimates as response variables given the problems we have attributed to them. However, as we discuss below, we only used the estimates from portions of speech where the formants are well-defined and so the estimates should be reliable.

Because we were interested in estimating formant frequencies in voiced sections of speech where they are well-defined, we only retained frames for which $B_1 < 400$ Hz and $B_2 < 300$ Hz. These values were arrived at by inspecting a scatter plot of B_1 and B_2 (Fig. 4.1) and noting that the density of points dropped off greatly outside the rectangle defined by these values. We inferred

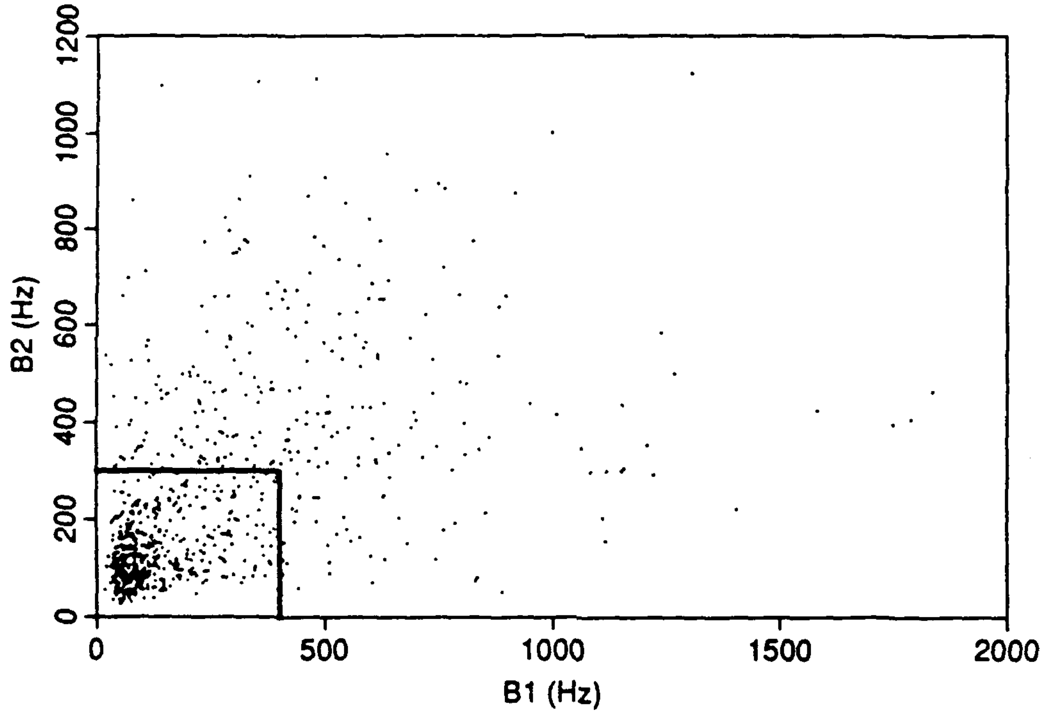


Figure 4.1: Formant bandwidths. B1 and B2 are bandwidths of first two formants estimated by ESPS formant tracker on 115 TIMIT utterances from 23 male speakers. Frames whose formant bandwidths fall within the rectangle are used to determine formant estimate regression coefficients.

from this that voiced speech tended to fall within the rectangle and unvoiced speech outside. In retrospect, these values seem very high for vowels in light of acoustic theory [Stevens 87]. Nonetheless, we used them in our work.

About 60% of frames had values of B_1 and B_2 that fell within the rectangle. The final training set included 1835 data points. We computed for each data point in the training set its formant frequencies and fifteen MFSPC regressors. Since the MFS frame advance was 5 ms and the formant tracker's was 10 ms, the regressors for each data point consisted of the MFSPC's averaged over the two frames coincident with the tracker frame.

Because it would have been time consuming, we did not check the estimates made by the formant tracker. Since only formants with relatively sharp resonances were used, we believe that the formant tracker estimates were reliable. In the following, we will refer to the formant tracker estimates as the "true" values F_k of the formant frequencies to distinguish them from the predicted values $\hat{F}_k$ computed from Eq. 4.1, although it should be kept in mind that both values are really estimates.

We used two methods to compare performance among the different regression models investigated. In the preliminary stage of model building we used the R^2 value for the fit, which measures the fraction of variance in the response variable accounted for by the model. It does not measure the ability of the model to predict the response variable on new data. This is particularly a problem in cases where there are few training data and many regression coefficients to estimate. In these cases, the model might be *overspecified* to the training data, yielding high values of R^2 but large prediction errors on new data. To deal with this problem, we used a test set of the five "sx" sentences from each of six male TIMIT speakers, none of whom were used in the training set. A total of 571 data points were collected from this test set in the same fashion as described above. From these data, the root mean squared

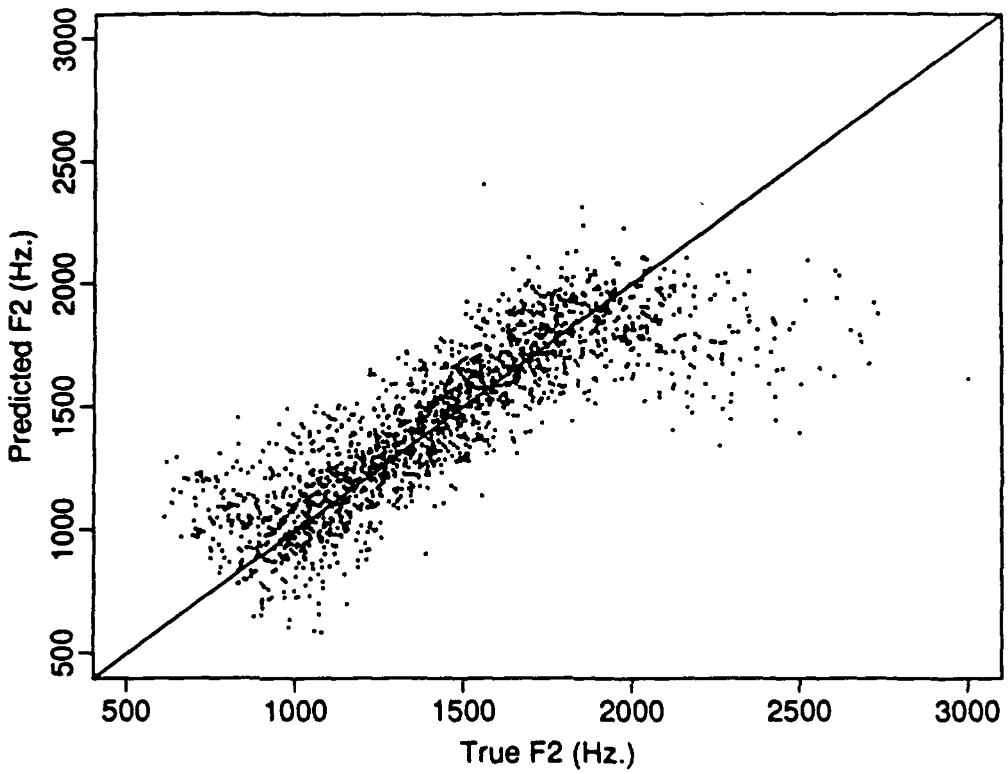


Figure 4.2: Fit of MFSPC-based regression model for F_2. Model determined using fifteen mel-frequency spectral principal components (MFSPC's) as regressors. Line represents $\hat{F}_2 = F_2$.

error (RMSE)

$$\sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}(F_{2i} - \hat{F}_{2i})^2}$$

where n_t is the number of points in the test set, is used as a metric for comparing model performance. We also report the correlation coefficient, r, between predicted and true values of F_2 in the test set [Myers 86, p.40].

A plot of F_2 predicted from the fifteen MFSPC's against the true F_2 for training set is shown in Figure 4.2. We can see from the plot that the fit is quite linear for moderate values of F_2 but not at low and high values. In particular, F_2 is overpredicted at low values and underpredicted at high values.

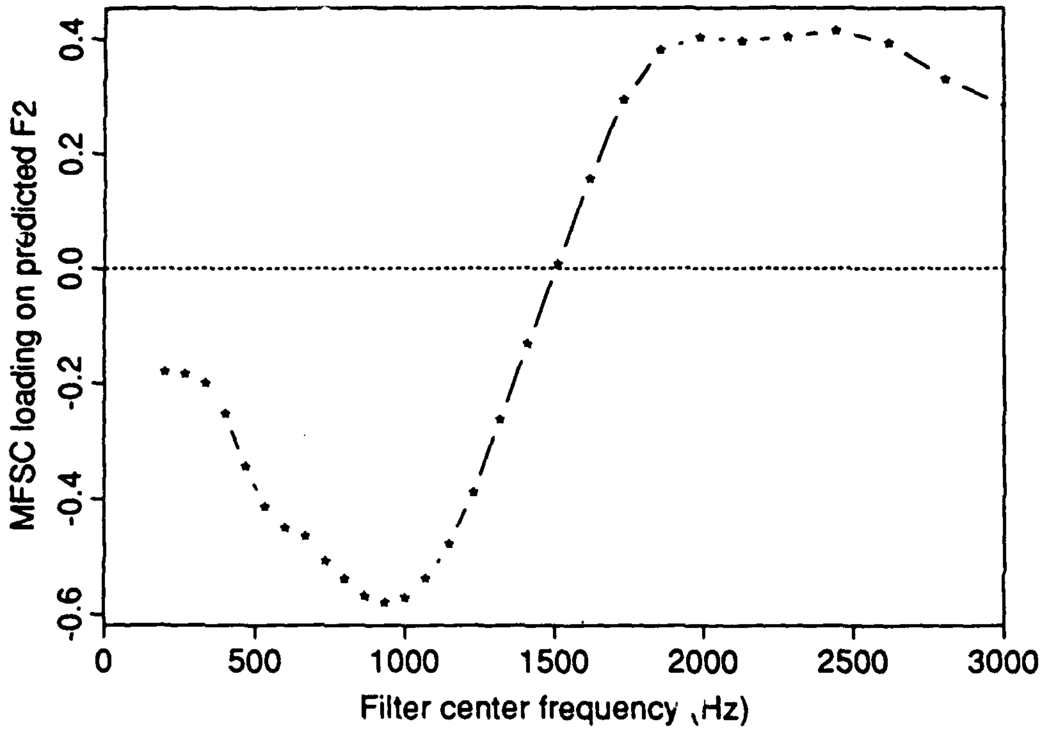


Figure 4.3: Loadings of MFSC's on predicted F_2. Each loading is the estimated correlation coefficient (r) between the corresponding MFS coefficient and the F_2 predicted from the MFSPC regression model. The loadings are plotted against the center frequency of the corresponding MFS filter. For clarity, only the frequency range from 0 to 3000 Hz is plotted.

To understand the behavior of the model, we computed correlations between the predicted F_2 values and the MFSC's as estimated from the regression model training data. These are referred to as "loadings" in the statistical literature, e.g., [Dillon 84, p. 31], and are useful for interpreting regression coefficients. Note that the regression coefficients were determined for the principal components of the MFSC's, not the MFSC's themselves. However, the loadings for the MFSC's on F_2 are meaningful in themselves and are easier to interpret than those on the principal components. The loadings were plotted against the center frequencies of the mel-frequency filters, as shown in Figure 4.3. We only show the range from 0-3000 Hz. Beyond this range, the curves tend towards a zero loading.

Where there is a large difference between loadings of adjacent filters, the predictor is most sensitive to relative values of the MFSC's in those filters. For example, there is a large difference in loadings of the filters near 1500 and near 1600 Hz. Thus, a data point for which the 1600 Hz MFSC is higher will tend to be associated with a higher predicted value of F_2 than in a case where the opposite is true. In general, at frequencies where the curve has greatest slope, the predictor is most sensitive to differences among MFSC's. Thus, the predictor appears to be most sensitive between 1000 and 1900 Hz. In effect, the predictor computes a quantity related to the a spectral "center of gravity" in the range of 1000-2000 Hz since its loading on each filter in that range is proportional to the frequency's distance from the center of the range, 1500 Hz. However, it is not sensitive at frequencies from 800 to 1000 Hz or from 1900 to 2500 Hz, indicating that differences in F_2 within these ranges will not be reflected by the predictor. This is consistent with the observed behavior of the predictor.

The range where the fit is good is probably limited due to fact that spectral prominences below 1000 Hz are most often associated with F_1 and those above 1900 Hz are often associated with F_3. Assume, for instance, that for a point in the regression training set, F_1 fell in the range of the predictor's sensitivity and had a more prominent spectral peak than F_2. The predicted F_2 value would then be the observed F_1 value, leading to a large prediction error for this data point. The least-squares method acts to avoid these large errors and so the presence of such data points would lead the method to compute the center of gravity over a smaller range. Thus, the method sacrifices prediction accuracy beyond the intermediate range for accuracy within that range, where most of the data points lie.

We thus reasoned that it would be difficult to predict values of F_2 beyond the midrange without building a more complex model that took the effect of other formants into account. Instead, we opted to concentrate on improving the model for midrange values of F_2 by removing from the regression training set points whose F_2 values fell outside the midrange. Thus, we traded off the accuracy of the the fit outside of the midrange, which was not very good anyway, for more accuracy within the midrange. We decided on an appropriate midrange by iteratively reducing the range to be predicted and making plots such as that in Fig. 4.2 until the fit looked good throughout the range. We finally settled on a midrange of 900 to 1900 Hz. covering 1473 (80%) of the F_2 values in the training set. The reported MMSE and r values of this and all subsequent models on the test set pertains to F_2 in the midrange.

In building a model to predict midrange F_2, we found that the MSE $\frac{1}{n}\sum_{i=1}^{n}(F_2 - \hat{F}_2)^2$ was slightly smaller when $\log F_2$ was used instead of F_2 as the response variable. Thus, the model is of the form

$$\widehat{\log F_2} = a_0 + \sum_{i=1}^{15} a_i c_i \qquad (4.2)$$

where c_i is the ith MFSPC. For this model, $\hat{F}_2 = \exp(\widehat{\log F_2})$.

Fig. 4.4 illustrates a plot of the midrange F_2 from the training set against the value predicted using Eq. 4.2. While there are slight deviations from linearity near 900 and 1900 Hz, they are not as severe as those for the model of the full F_2 range, as displayed in Figure 4.2. Also, there is less spread around the line $F_2 = \hat{F}_2$ for the midrange model. The RMSE for this model

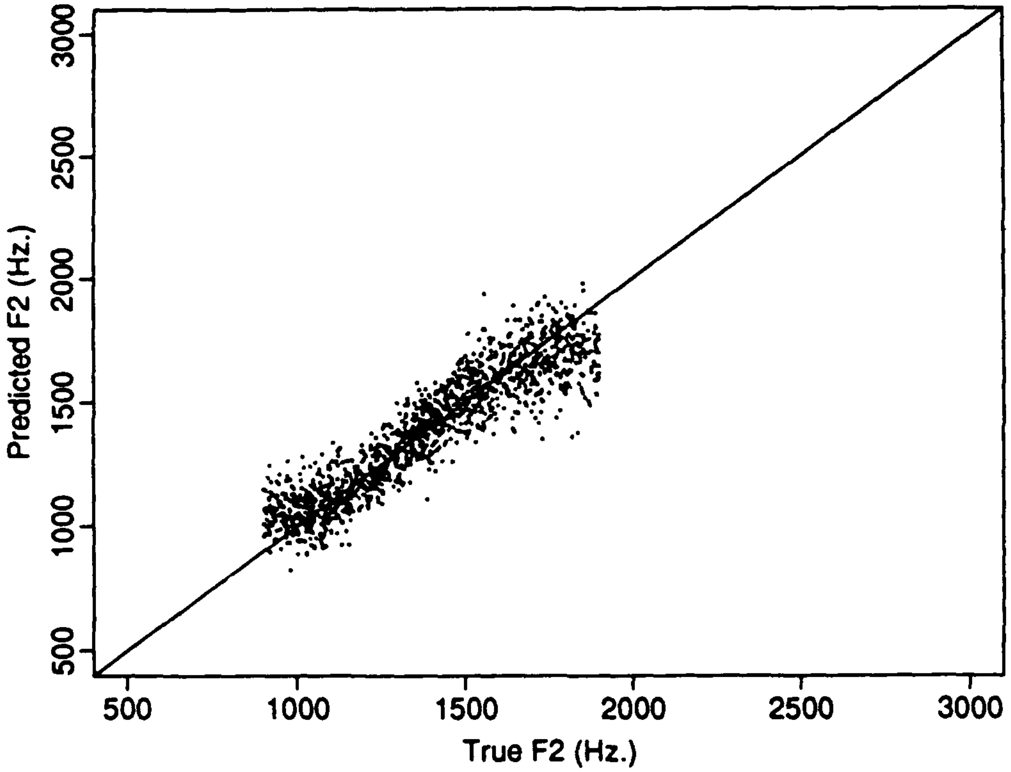


Figure 4.4: Fit of MFSPC-based regression model for midrange F_2. Model determined using fifteen mel-frequency spectral principal components (MFSPC's) as regressors and $\log F_2$ as the response variable. Line represents $\hat{F}_2 = F_2$. Scale same as in Figure 4.2 to facilitate comparison.

on the test set F_2 is 103 Hz. and $r = .934$. These values were similar to those obtained on the training set, verifying that the model was not overspecified.

While this is a moderately high value of r, we believed that the model could be improved by exploiting the concept of a center of gravity computation more fully. We were inspired in this by work described in [Zue 89b] which provided anecdotal evidence that the center of gravity computed using the hair-cell envelope spectral representation tracked formant frequencies quite well. As we show below, the regression model of Eq. 4.2 is incapable of computing a true center of gravity.

To be specific, let us introduce a center of gravity computation based on the set of MFSC's $e_j, 1 \leq j \leq 40$. Let ℓ and h be the index of the lowest and highest frequency coefficients used and let f_j be the center frequency of the jth filter used to represent the mel-frequency spectrum. Finally, let $C(\ell, h)$ be the center of gravity determined for a specific (ℓ, h) pair. Then

$$C(\ell, h) = \frac{\sum_{j=\ell}^{h} f_j e_j}{\sum_{j=\ell}^{h} e_j}. \tag{4.3}$$

The true center of gravity $C(\ell, h)$ can be used directly to estimate F_2. However, a better estimate can probably be derived by using the multiple regression framework to solve the problem. Formally, let

$$g_j(\ell, h) = \frac{e_j}{\sum_{j=\ell}^{h} e_j}, \qquad \ell \leq j \leq h. \tag{4.4}$$

We will rєfer to the g_j as *center of gravity (CG) coefficients*. Then the equation

$$\widehat{\log F_2} = a_0 + \sum_{j=\ell}^{h-1} a_j g_j(\ell, h) \tag{4.5}$$

can be used as the estimate. As usual, thc set of a_j denote the regression coefficients. The set of $g_j(\ell, h)$ are the regressors. The reason that the largest index j is $h-1$ and not h is that $\sum_{j=\ell}^{h} g_j = 1$ so that only $h - \ell$ out of $h - \ell + 1$ elements of the set of g_j are linearly independent. Thus, one of the regressors can be eliminated from the regression model and we arbitrarily eliminated g_h.

Ignoring the fact that $\log F_2$ is being estimated instead of F_2, this equation is similar to Eq. 4.3 except that the coefficients a_j are determined by regression instead of being set to the center frequencies f_j. Thus, the model is capable of computing the true center of gravity defined in Eq. 4.3 if this corresponds to the minimum MSE estimate of $\log F_2$. The transformation from the set of e_j to the set of g_j is non-linear. Since the principal component transformation from the MFSC's to the MFSPC's is linear, Eq. 4.2 describes a linear combination of the MFSC's and thus cannot be used to compute a center of gravity.

To use Eq. 4.5, ℓ and h must be specified. Because F_2 was being estimated over the range of 900-1900 Hz., we tested models for a number of (ℓ, h) pairs for which f_ℓ and f_h were near 900 and 1900 Hz, respectively. No model yielded a substantial decrease in RMSE compared to the MFSPC regressor model of Eq. 4.2.

We hypothesized that this disappointing result could have been due to the limited dynamic range of the MFSC's. From Eq. 4.3 one can see that each frequency f_j in the center of gravity computation is weighted by e_j, the corresponding MFSC. If there is not a great difference among the weights, no frequency will dominate the computation and the center of gravity computation might not be sensitive to a spectral peak. This applies to Eq. 4.5 as well, since it is based on Eq. 4.3.

To test this hypothesis, we generalized the center of gravity computation of Eq. 4.3 to

$$C(\ell, h, d) = \frac{\sum_{j=\ell}^{h} f_j \exp(de_j)}{\sum_{j=\ell}^{h} \exp(de_j)} \tag{4.6}$$

where d is a parameter that controls the dynamic range of the coefficients. Note that when $d = 0$, the equation is identical to Eq. 4.3. When $d = \infty$, the center of gravity is the center frequency corresponding to filter whose MFSC is highest. Finally, when $d = 1$, the weights in the center of gravity are the exponentials of the MFSC's. The CG coefficients g_j corresponding to this generalization are expressed as

$$g_j(\ell, h, d) = \frac{\exp(de_j)}{\sum_{j=\ell}^{h} \exp(de_j)}, \qquad \ell \le j \le h. \tag{4.7}$$

We tested models using the generalized regressors for various (ℓ, h, d) triples. The lowest test set RMSE of 47 Hz. was obtained for $\ell = 13$ ($f_\ell = 1000$ Hz.), $h = 22$ ($f_h = 1850$ Hz.), and $d = .75$. Thus, the RMSE for this model was less than 50% of that of the model based on MFSPC regressors. The correlation coefficient for this model was .983.

Finally, we combined the two sets of regressors into the model

$$\log \hat{F}_2 = a_0 + \sum_{i=1}^{15} a_i c_i + \sum_{j=13}^{21} b_j g_j(13, 22, .75) \tag{4.8}$$

where a_i and b_j are regressor coefficients. This yielded an RMSE of 45 Hz. and $r = .985$ on the test set, a slight improvement over the previous model. Again, the corresponding measures of performance on the training set were similar. Thus, even though this model has a greater number of regression coefficients to determine, it is not overspecified. Since MFS filter spacing in the range of 900 to 1900 Hz varies from about 70 to 130 Hz., it is probably difficult to do better than this.

A plot of the predicted vs. true values of F_2 on the test set for this model is shown in Fig. 4.5. While for almost all cases the model performs well, there are some errors of 200 Hz or more. We hypothesize that they may be due to the presence of zeroes in the vocal tract transfer function that split the formants into two peaks. This might lead to the frequency at one of the peaks rather then a value between the two peaks being predicted as the formant.

The process of building a model for F_1 was similar. The model was based on values of F_1 below 700 Hz., comprising 92% of the training data. We will refer to this range as *low range* F_1. Again, the best regressor set consisted of a combination of the MFSPC and CG's and $\log F_1$ was used as the response variable. For the best model, the RMSE is 26 Hz. and $r = .973$.. This was

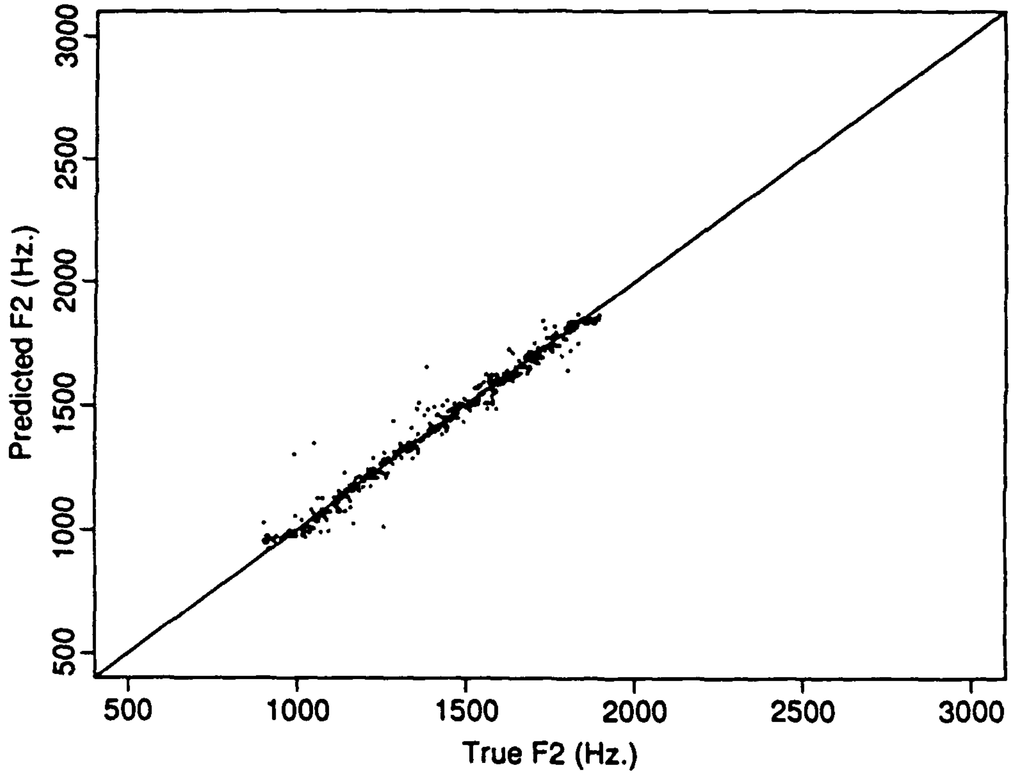


Figure 4.5: Fit on test set of best regression model for midrange F_2. Model uses fifteen mel-frequency spectral principal components (MFSPC's) and ten center of gravity (CG) coefficients as regressors. The model's response variable is $\log F_2$. Line represents $\hat{F}_2 = F_2$.

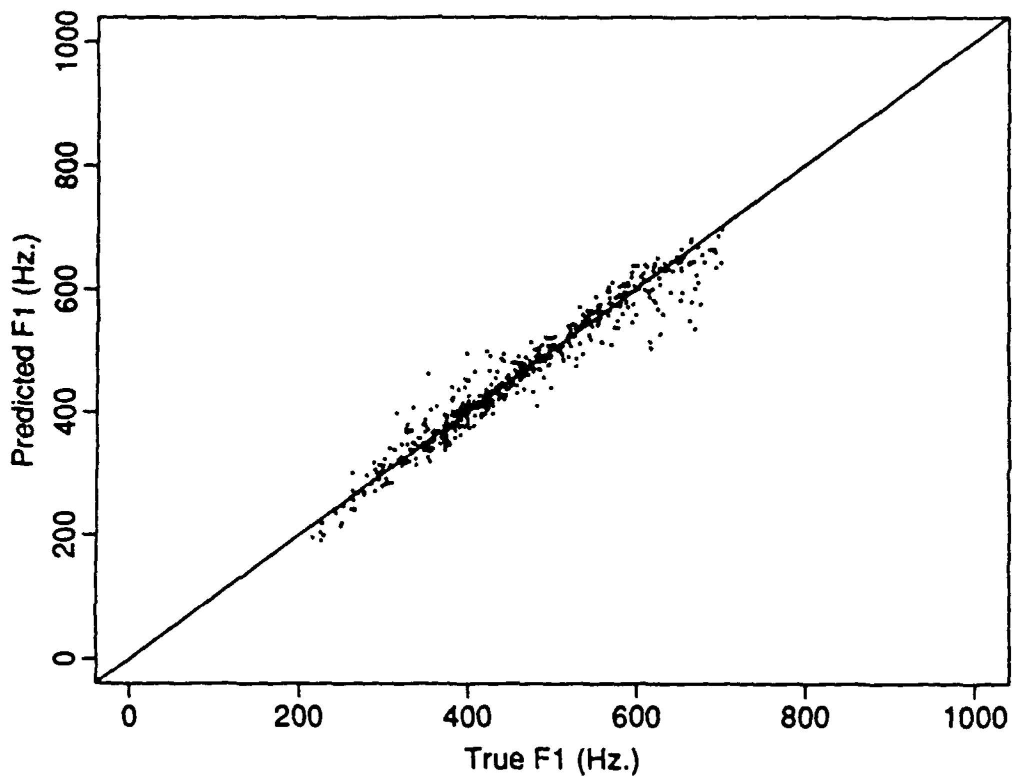


Figure 4.6: Fit on test set of MFSPC-based regression model for low range F_1. Model determined using MFSPC's and CG regressors as discussed in text. Line represents $\hat{F}_1 = F_1$.

obtained with $\ell = 1$ ($f_\ell = 200$ Hz.), $h = 8$($f_h = 670$ Hz.) and $s = .25$. A plot of the predicted vs. true F_1 for the test set is shown in Fig. 4.6.

The filter separation in the 200-700 Hz. range is a constant 67 Hz. Thus, it is unlikely that any model could obtain an RMSE substantially below 26 Hz. We do not have any explanation for the different optimal values of d obtained for predicting F_2 and F_1. In any case, performances of both models were relatively insensitive to choices of d in the range of .25-.75. Table 4.1 compares the performance of the models discussed in this section for predicting midrange F_2 and low range F_1.

	F_1 Model Performance		F_2 Model Performance	
	MSE (Hz.)	r	MSE (Hz.)	r
MFSPC's	40	.931	103	.934
CG's	33	.961	47	.983
Both	26	.973	45	.985

Table 4.1: Performance comparison of regression models for predicting midrange F_1 and F_2. The regressors for the three cases are: (1) the fifteen mel-frequency spectrum principal components (MFSPC's), (2) the center of gravity coefficients used for each formant (CG's) and (3) both. The model's explanatory variable in all cases was the logarithm of the formant frequency. Both correlation (r) and mean square error (MSE) reported for test set.

We attempted to build a regression model for F_3 but were unable to get good results. Thus, we only included estimates of F_1 and F_2 in the phonetic recognition experiments described in Section 4.8.3. We will refer to these estimates as predicted formants and abbreviate them as PF_1 and PF_2.

4.4.3 Comparison to Previous Work

Broad and Clermont [Broad 89] built multiple regression models of F_1, F_2 and F_3 using 14 LPC-based cepstral coefficients as regressors. Unlike our study, formants were hand-edited, and moving averages of both the coefficients and the formants were used.

They performed a pilot study for a single male speaker and a more comprehensive study with a set of four male speakers. For the pilot study, plots of true vs. predicted formants on the training set did not exhibit the poor fit outside the midrange seen in Fig. 4.2. The poorer fit of our model may be due to the fact that data were collected from several speakers and from all portions of speech assumed to be voiced rather then from vowels alone. However, there are at least two other plausible explanations for the difference:

1. A regression model based on the LPC cepstrum might somehow avoid confusions among formants. The authors do report that errors with the

LPC cepstrum are smaller than those obtained using a cepstrum directly computed from the log spectrum. However, they do not compare LPC and mel-frequency cepstra.

2. The model is overspecified so that a plot of the fit on new data would exhibit the aforementioned behavior. This is likely given that in the pilot study, the 660 points used to train the model are collected from only 180 tokens and only 60 distinct syllables. In fact, in the comprehensive study the authors do indeed report that speaker-specific prediction errors on new data are a factor of 1.6 greater than those on training data. The comprehensive study collected more data per speaker than the pilot study did. Thus, it is likely that the pilot study suffered even more from overspecification.

The results in [Broad 89] most relevant to our study concern the model performance for multiple speakers on a test set distinct from the training set. For this condition, the reported RMSE for F_1 and F_2 are 52 Hz and 164 Hz, respectively. These are much larger than the errors obtained by our models. A key reason for this, of course, is that our models attempt to predict formants over a smaller range so a comparison between the two models is not really fair. Other reasons may include our use of more training data and the non-linear transformation to CG coefficients that we use in our regressor set.

Other related work includes a series of studies based on the relationship between F_1 anJ F_2 in Dutch vowels and log spectral energies of a bank of 18 wideband filters. The work was reported in [Pols 69, Klein 70, Pols 73]. In [Pols 69], high correlations were found between formants and linear combinations of the first six principal components of the filter outputs for vowels from a single speaker. In [Klein 70], formants and filter outputs for twelve vowels were averaged data from fifty male speakers. High correlations was determined between the first four principal components of the averaged vowel spectra and the averaged log F_1 and log F_2. In [Pols 73] a similar result was obtained using linear combinations of filters determined by discriminatory criteria rather then principal components. The act of averaging the data is likely responsible for the high accuracy of linear models reported in these papers compared to ours.

4.4.4 Discussion

The major substantive result of this section is that a linear model is inadequate for characterizing the relationship between the MFSC's and formants. By understanding the mechanism behind the relationship we were able to develop a better model, albeit over a limited range. However, we have not been able to determine whether the addition of formant estimates to MFSC's can improve recognition performance. The lack of substantial improvement when they are added, as demonstrated in Section 4.8.3, may be due instead to the limited range of the models. This is discussed further where we present the results. Thus, it would be a worthwhile extension to our work to improve the models as this might settle the question.

Another possible extension would be to apply the process described in this section to modeling other features that are known to be useful for discrimination. For instance, it might be interesting to model the voice onset time (VOT) for a stop consonant with the recognizer's measurement set since VOT is known to be a good cue for distinguishing voiced from unvoiced consonants [Lisker 64]. The failure to find a strong relationship between the measurement set and the VOT coupled with a tendency of the recognizer to make many voiced/voiceless confusions might indicate a problem that needs addressing. We discuss speech recognition diagnostics in depth in Chapter 6.

4.5 Out-of-Segment Measurements

We have suggested that an important property of our segmenter is that segment boundaries may be useful places about which to make measurements useful for discrimination. To take advantage of this property, we included in our study measurements made just outside the segments being modelled. We refer these as *out-of-segment* measurements. In particular, we focused our attention on measurements that would be useful for discriminating voiced stops. The place of articulation of stop consonants can often be identified from the direction of formant movements and values of formant frequencies in vowels just following the stop burst and just preceding the stop closure (if one exists) [Stevens 78, Sussman 91]. Thus, formant frequencies measured just outside those associated with stops were good candidates for the study. In particular, we focused our attention on voiced stops. Formant movements are of particular importance for discriminating them because they are often unaspirated. Thus, the aspiration spectrum, which can be used to distinguish voiceless stops, is of less use in distinguishing voiced ones.

To determine measurement locations before and after each segment, we measured discrimination among stop closures and bursts, respectively. We will describe the procedure used to determine measurement locations after the burst segments. An identical procedure was used to determine the locations before the closures.

For each segment sequence in the acoustic model training set that was associated with a stop burst, we computed PF_1 and PF_2, the formants predicted from the model of the last section, for each frame from one to ten frames (5-50 ms) beyond the right edge of the final segment in the sequence. We did not check the identity of the following phonetic label to the right but we assume that in most cases the stops were followed by vowels. Thus, for most segment sequences the measurements were presumably made at the onset of voicing.

Let the value of the predicted formant PF_k, $k = 1$ or 2, measured j frames beyond the right edge of the segment be PF_{kj}. For each token of the voiced consonants /b/, /d/, and /g/, and each frame pair u and v we measured formant movements

$$\Delta_k(u,v) = \mathrm{PF}_{kv} - \mathrm{PF}_{ku}, \quad u \leq 1 \leq v \leq 10.$$

for each formant. Finally, for each formant and each pair of voiceless consonants we computed the magnitude of the difference in class means of the formant movements $\Delta_k(u, v)$ normalized by an estimate of the within-class standard deviation pooled over the two classes. This is a the one-dimensional version of a widely used measure of *class separation* [Duda 73,p.29].

We then plotted this measure of separation as a function of u and v using perspective plots to determine the (u, v) pairs that provided the most class separation. These plots are shown in Fig. 4.7. In the top right plot, for example, it can be seen that when PF_2 is being used to discriminate between /b/ and /d/, the maximum class separation occurs when the measurements are made 5 ms and 35-50 ms into the following segment. By inspecting for each pair of classes the perspective plot for the formant providing the best separation (PF_1 for /b/-/g/ and PF_2 for the others), we concluded that measurements made 5 and 35 ms after the segment provide close to the best separation for all three pairs.

To check that the formant movements were in the direction we expected, we generated for each voiced stop boxplots [Chambers 83] of the differences between measured at 5 ms and 35 ms after the segment. They are shown in Fig. 4.8. The horizontal line segment in the middle of each box represents the median, the box extends from the first through third quartiles and the bars extending out of the box are each of length 1.5 × interquartile range.

While the plots indicate large spreads in the distributions, there is a general correspondence between the observed results and those that would be predicted from prior acoustic-phonetic knowledge. For instance, the plots of PF_1 indicate that formants tend to rise out of all three consonants, but most for /g/. The rise is in accord with acoustic-phonetic theory [Kewley-Port 82] although we are unaware of any theory behind the greater rise for /g/. Also, as expected, the PF_2 difference tends to be positive for /b/, reflecting rising formant movement due to the labial place of articulation. That of /d/ tends to be near 0 or negative since /d/ often pulls F_2 up towards 1800 Hz

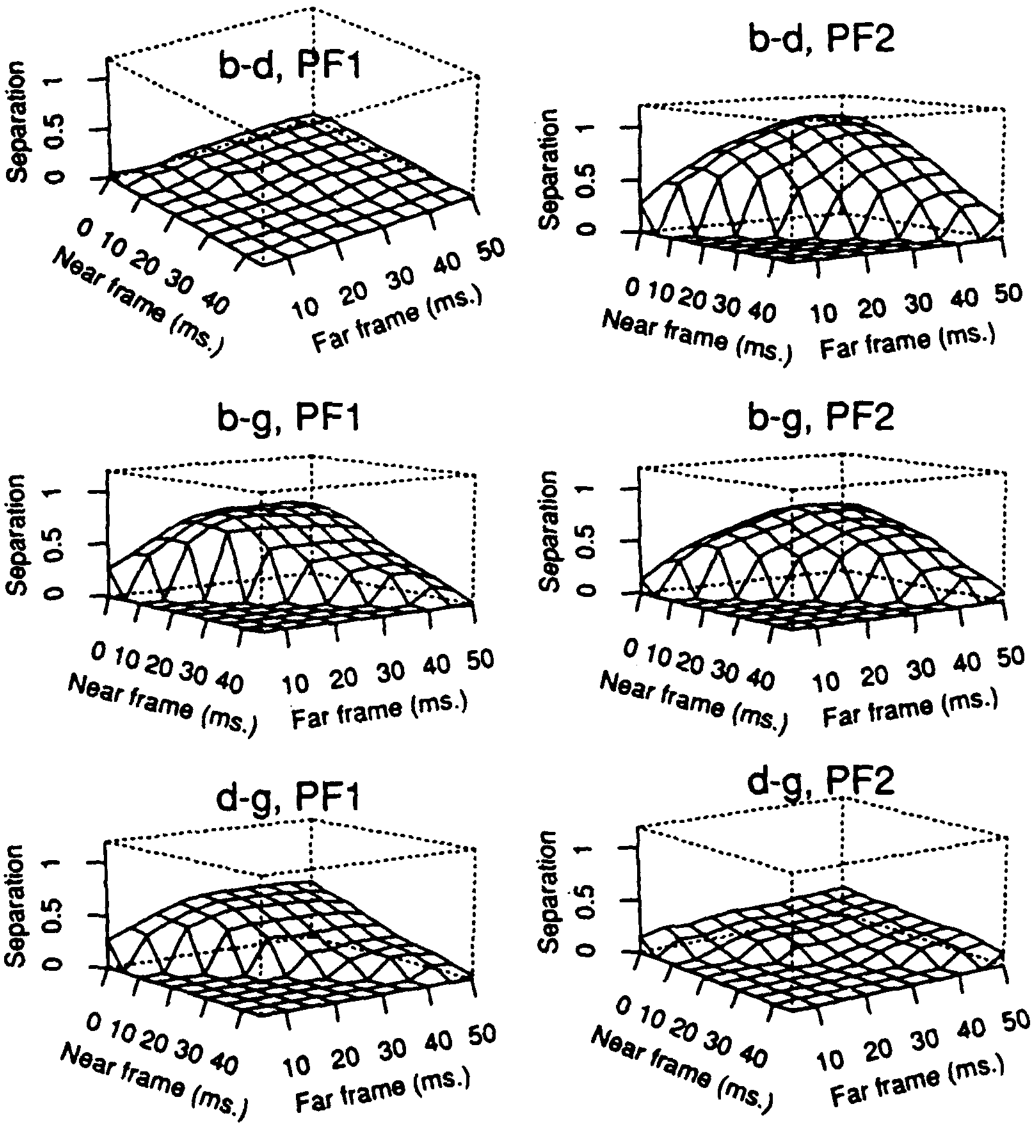


Figure 4.7: Voiced stop separation as a function of formant measurement location. For each pair of classes, the standardized difference in class means of formant change between the two locations where the formant is measured is used as the measure of separation. The near and far frame axes indicate the intervals between the end of the segment and the earlier and later frames at which the formant is measured, respectively. The class pair and predicted formant identifier appear above each plot.

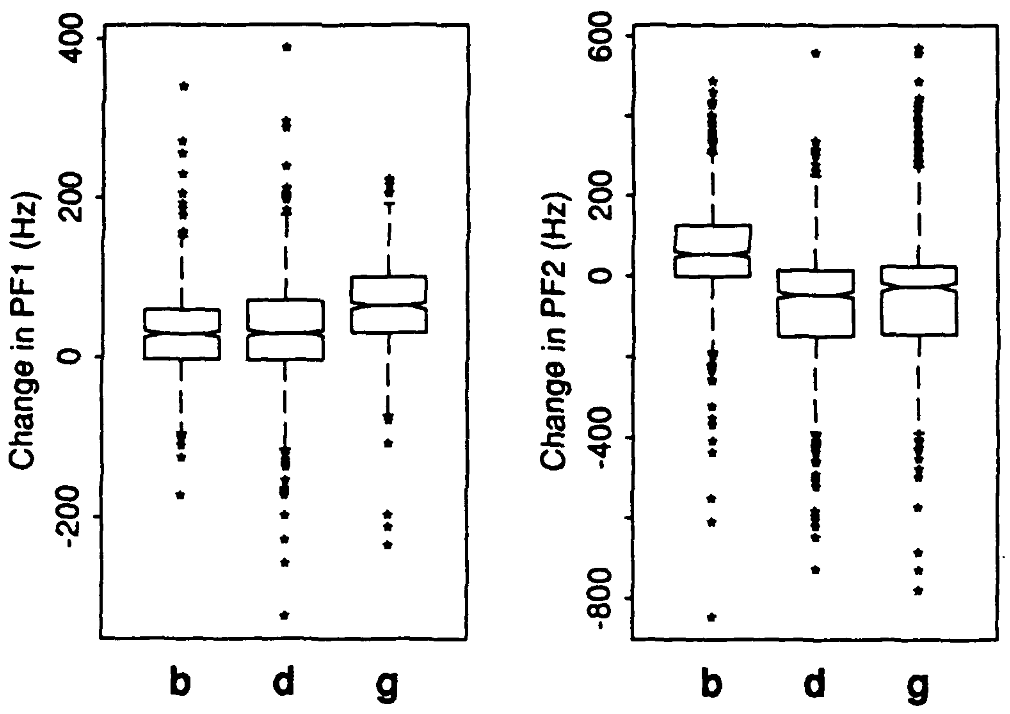


Figure 4.8: Formant movements between 5 ms and 35 ms after segment for voiced stops. Boxplots are as described in [Chambers 83]. The horizontal line segment in the middle of each box represents the median, the box extends from the first through third quartiles and the bars extending out of the box are each of length 1.5 x interquartile range. Extent of notches indicates 95% confidence interval in mean of the change in formant frequency. Left boxplot shows change in PF_1 and right shows change in PF_2. The identity of the voiced stop appears under each boxplot.

[DeLattre 55]. The same is true of /g/, which tends to pull F_2 up towards F_3. The correspondence of these results to their predicted behavior is evidence that the segmenter is behaving in a reasonably consistent manner following voiced stops. Additionally, the class separations, while not great, are large enough to suggest that the predicted formants are useful measurements for performing phonetic discrimination.

Results obtained by measuring before the stop closures were quite symmetric with these, so we retained spectral measurements made at 5 and 35 ms before the start of each segment. The perspective plots illustrating separation among the closures is shown in Fig. 4.9.

Thus, the results of these experiments led us to include spectral representations measured at 5 and 35 ms before and after the segment edges. Each spectral representation includes the 15 MFSPC's, the two predicted formants and the energy. We will abbreviate all measurements made 5 and 35 ms after the segment using the suffixes "F" (for "following") and "FF" (for "following following"), respectively. Similarly, measurements made before the beginning of the segment will have prefixes "P" and "PP" for "preceding" and "preceding preceding." Thus, for example, the spectral representation of the frame 35 ms after the end of the segment includes the 18 measurements $\mathrm{FFMFSPC}_i$, $1 \leq i \leq 15$, FFPF_1, FFPF_2, and the energy measurement FFENGY.

Including out-of-segment measurements may lead to a greater degree of statistical dependence among segment observation vectors since the same spectra may be included in the observation vectors of distinct segments. However, we chose to ignore this problem in our work.

4.6 Summary of Measurements

Figure 4.10 depicts, for a single segment, all the measurement subsets discussed in the chapter. The complete set of measurements includes eight spectral representations. Their locations and the abbreviations used to refer to

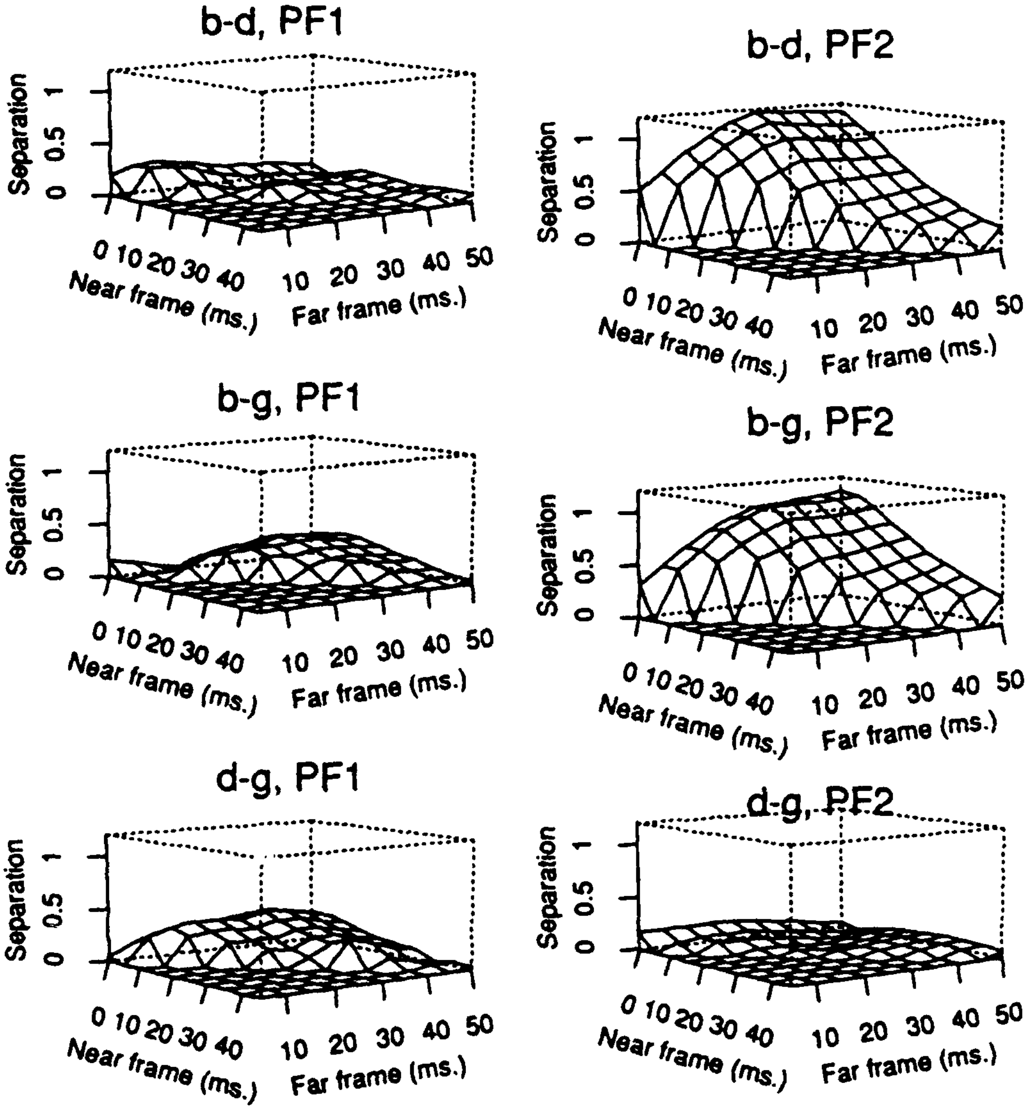


Figure 4.9: Voiced stop closure separation as a function of formant measurement location. For each pair of classes, the standardized difference in class means of formant change between the two locations where the formant is measured is used as the measure of separation. The near and far frame axes indicate the intervals between the beginning of the segment and the later and earlier frames at which the formant is measured, respectively. The class pair and predicted formant identifier appear above each plot.

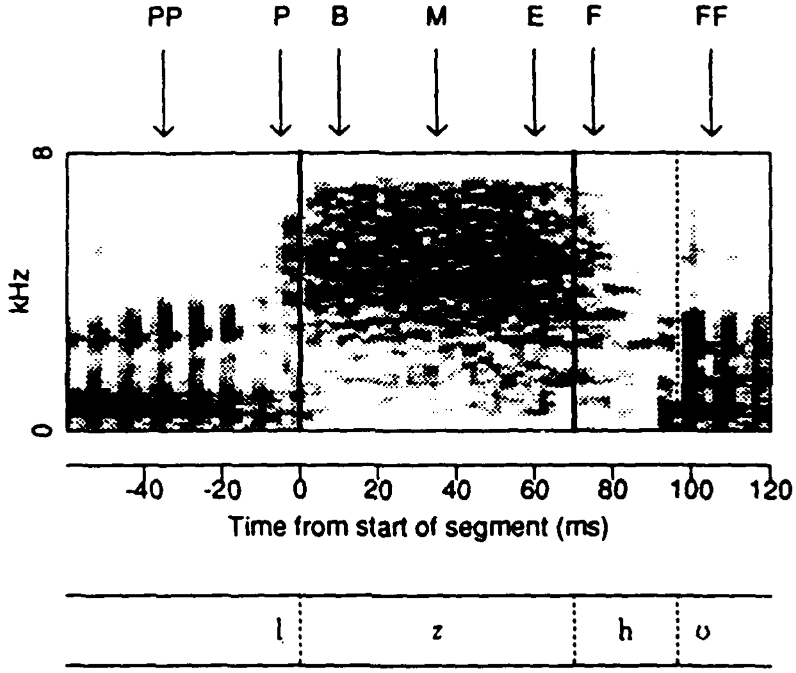


Figure 4.10: The complete measurement set. Arrows point to positions where spectra are measured for each segment. Letters associated with arrows are the position abbreviations. Average spectrum and duration not depicted. (PP: *preceding-preceding*, P: *preceding*, B: *beginning*, M: *middle*, F: *following*, FF:*following-following*)

them are listed in Table 4.2. The measurements made at each position are listed in Table 4.3. Note that there are eight spectral positions and 18 measurements made at each position. Each of these measurements is abbreviated by concatenating its position and measurement type abbreviations. To refer to the beginning predicted first formant, for example, we use the abbreviation BPF_1. The complete set of segment measurements considered in the phonetic recognition experiments also includes segmental duration (abbreviated DUR). Thus, there a total of $8 \times 18 + 1 = 145$ in the complete set.

Abbreviation	Measurement location
PP	35 ms before segment start
P	5 ms before segment start
B	10 ms after segment start
M	segment middle frame
E	10 ms before segment end
F	5 ms after segment end
FF	35 ms after segment end
A	average over segment

Table 4.2: Positions of mel-spectral-based measurements.

Abbreviation	Measurement	Number
MFSPC	Mel-spectral coefficient principal components	15
PF	Predicted formants	2
ENGY	Energy estimate	1

Table 4.3: List of spectral measurements.

4.7 Multiple and Grouped Multiple Discriminant Analysis

4.7.1 Introduction

Multiple discriminant analysis (MDA) is a well-known technique for reducing dimensionality in classification problems. Given an $n \times p$ data matrix $\boldsymbol{X}$ of measurement vectors of length p drawn from a set of k classes, MDA computes a set of transformation vectors $\boldsymbol{r}_j, 1 \leq j \leq \min(p, k-1)$. The variable $\boldsymbol{X}\boldsymbol{r}_1$ has the highest ratio D of between-class to pooled within-class sample variance attainable by any linear transformation.[1] Thus, this variable is likely to be useful for discriminating among the classes. The variable $\boldsymbol{X}\boldsymbol{r}_2$ has the highest D among all variables uncorrelated with $\boldsymbol{X}\boldsymbol{r}_1$ and so on. In general, given some measurement vector $\boldsymbol{x}^T$, the value $\boldsymbol{x}^T\boldsymbol{r}_j$ is referred to as the j^{th} *linear discriminant* for that vector. The technique is described in detail in [Duda 73, Johnson 88].

As with principal components analysis, MDA computes a set of linear transformations of the measurement set and orders them according to some criterion. The advantage of MDA over principal components analysis is that it considers discrimination ability when ordering the transformed variables, while principal components analysis does not. In particular, it is possible that the set of q principal components with the highest variances are not the set of q transformed variables most useful for phonetic discrimination. On the other hand, the first q discriminants can be shown to provide the minimum classifier error rate of any set of q transformed variables under the assumptions that the class PDF's are Gaussian and that each has the same within-class covariance [Johnson 88, p. 534]. Even if these conditions are not met, D is a

[1] In general, if there are k classes, and each class i has n_i data points associated with it and a within-class sample covariance matrix of S_i then the pooled within-class covariance matrix S_{pooled} is $\sum_{i=1}^{k} n_i S_i / \sum_{i=1}^{k} n_i$. The pooled within-class sample variance $\sigma_{\text{pooled}}{}^2$ has a similar form but in one dimension. The pooled within-class standard deviation, which is referred to in Section 4.7.3, is thus σ_{pooled}.

more reasonable criterion than variance for determining transformations useful for discrimination.

Pols [Pols 73] discussed the use of MDA as a technique for determining measurements useful for vowel classification and pointed out its theoretical superiority over principal components analysis for classification. Brown [Brown 87] provided a detailed description of the technique, used it in a recognition task and showed that a recognizer using a given number of linear discriminants could outperform one using the same number of principal components. The technique has also been used in speech recognition by Hunt [Hunt 89] and others.

4.7.2 Implementation

In a speech recognition system, the transformation determined by MDA is used to transform a measurement vector to an observation vector of q discriminants. The set of observation vectors so produced are use to train the subword models. In the recognition process, each measurement vector in the utterance to be recognized undergoes the same transformation into a vector of length q and the acoustic match of this vector to each model state is used in the scoring process.

To compute the discriminants, it is necessary to provide a set of training measurement vectors, each of the same length and each of which is labelled with the class to which it belongs. Thus, it is first necessary to define the set of classes to be discriminated. One possibility would be to consider each state in all the phone/biphone models to be a distinct class. Then all measurement vectors used to estimate the PDF for a particular state would be labelled with that state for the purposes of the analysis. However, the role of a recognizer is to distinguish labels, not states, from each other. Thus, it is inappropriate that states within a given model should be assigned different classes.

A more appropriate set of classes for phonetic recognition, at least, would be the set of phone labels. In a system where each segment is associated with a single phone token, the labelling problem would be straightforward. Each segment's measurement vector would be labelled with the associated phone. However, we had to adapt the method to our system, for which lexical labels include both phones and biphones and for which more than one segment can be associated with each token. We simplified the problem by only using segment sequences associated with single phones in the MDA training set. Thus, only the 57 classes associated with phones were used instead of the 141 associated with both phones and biphones. This simplification is reasonable in that it is not necessary to discriminate an /o/ that is associated with the lexical label **o** from one associated with the lexical label **o-r**, for example. Additionally, the bulk of segments are associated with the 57 phone labels so it is most useful to concentrate on discriminating them from each other.

The straightforward way of dealing with the issue of multiple-segment sequences being associated with some phone tokens would have been to include all segments in the MDA training set, each labelled with its associated phone label. However, this would have biased the training towards tokens associated with longer segment sequences since these tokens would contribute a disproportionate number of segments to the training set. Moreover, this tactic would have provided the same label to segments associated with different portions of a token, even though these segments might possess very different acoustic characteristics. For example, if a token of the diphthong /a^y/ is associated with two segments, the first will be quite different spectrally from the second and both would differ from a one-segment token of /a^y/. Thus, the spectrum measured at a particular place in a segment occupying one of these positions characterizes a different part of the phone than the spectrum measured at the same place in a segment occupying another position. For example, the BMF-SPC's measured at the beginning of the first /a^y/ segment would characterize the beginning of the diphthong while that measured at the beginning of the second segment would characterize the middle of it.

Instead, we represented each token with a single measurement vector com-

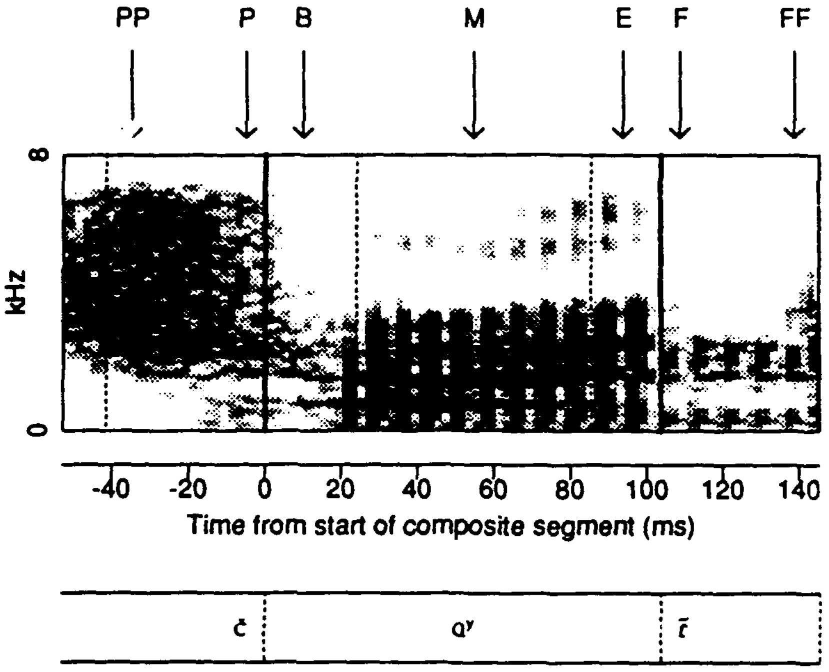


Figure 4.11: The complete composite segment measurement set. Arrows point to positions where spectra are measured to build a composite segment from a three-segment token. Letters associated with arrows are the position abbreviations. Average spectrum and duration are derived from the middle segment. They are not depicted. (PP: *preceding-preceding*, P: *preceding*, B: *beginning*, M: *middle*, E: *end*, F: *following*, FF:*following-following*)

posed out of of parts of measurement vectors from each segment in the sequence associated with the token. Figure 4.11 depicts the technique. Measurements characterizing the beginning (the preceding-preceding, preceding and beginning spectra) and end (the end, following, and FF spectra) of the token were taken from the first and last segments in each sequence, respectively. Those characterizing either the middle or complete segment (the middle and average spectra and the duration) were taken from the middle segment when the number of segments in the sequence was odd and from segment 1 +

number of segments/2 when the number was even. We will refer to the set of measurements as a *composite segment measurement set* or, for short, as a composite segment. We will refer to the within-class covariance matrices computed for each phone label from these segments as within-phone covariance matrices.

Composite segments are artifices in the sense that they do not correspond to actual segments, except in cases where there is a one-segment token. Thus, there is no theoretical justification for treating them as units to be discriminated. However, they can be rationalized as follows:

1. According to Table 3.5, 62% of the tokens of single phones are associated with a single segment so that in these cases, the composite segment does correspond to a single segment.

2. We have discussed the value of making measurements near segment boundaries since these correspond to acoustic landmarks. However, the segment boundaries between phonetic regions are probably more important acoustic landmarks for phonetic discrimination than those within the regions. For example, we justified the out-of-segment measurements in terms of their ability to characterize spectral change near stop-vowel boundaries. Because the measurements for characterizing the begin and end of the segment are made on the first and last segments in the sequence, the discriminant analysis is based on measurements made near these boundaries rather then on those made within the phonetic region.

For these reasons, we used composite segments in both the conventional and grouped multiple discriminant analyses. We introduce the latter technique next.

4.7.3 Grouped Multiple Discriminant Analysis

Rationale

The MDA computation uses the pooled covariance matrix, computed as a weighted average of the class-specific matrices. For this reason, MDA performs best at finding transformations useful for discrimination when the within-class covariances are nearly equal to each other and, as stated above, is the optimal dimensionality reduction technique under the assumption of Gaussian class-specific PDF's with equal covariance matrices.

Because of the widely differing speech sounds associated with the 57 phone labels, we doubted that the equal-covariance assumption was valid for the composite-segment within-phone covariance matrices used to perform MDA. However, a more reasonable assumption might be that covariance matrices within a group of related phones (e.g., the vowels) are similar. If this were the case, applying MDA *to the phones in the group* alone would likely yield more useful within-group discriminants than would be produced by applying MDA to the complete set of phones. If each of the 57 phones were assigned a group, within-group discriminants could be computed for each group, presumably improving overall within-group phonetic discrimination. This set of discriminants would not directly deal with the problem of between-group discrimination. However, if phones in different groups were very different acoustically, it would be unlikely that they would be confused given any reasonable set of transformations of the original measurement set. Thus, the set of within-group discriminants would potentially yield better overall phonetic discrimination than the set of discriminants determined from all the phone labels simultaneously.

In this subsection we develop a method for clustering covariance matrices. We used the method to cluster the composite-segment within-phone covariance matrices and performed multiple discriminant analyses on the resulting groups. We term this technique for computing discriminants *grouped multiple* *discriminant analysis* (GMDA). We show below that distinct clusters closely correspond to distinct manners of articulation. For one of the groups, that including vowels and semivowels, we show a correspondence among the first few discriminants and distinctive features. Finally, in Section 4.8.3, we compare phonetic recognition results obtained using GMDA and conventional MDA for reducing dimensionality.

Measurement Set used in GMDA Experiments

For our work with grouped multiple discriminant analysis, we used the following measurement set:

1. the complete B, M, E, and A spectral representations (72 measurements),
2. the ENGY, PF_1, PF_2 and $MFSPC_1$ measured in the PP, P, F, and FF positions (16 measurements),
3. differences between positions PP and P of MFSPC's 2 through 15 (14 measurements),
4. differences between positions F and FF of MFSPC's 2 through 15 (14 measurements), and
5. segment duration.

The set has a total of 117 measurements. We used the MFSPC differences in sets (3) and (4) rather then the spectral representations at positions PP, P, F and FF to reflect our belief that it is the spectral differences rather then some other function of the out-of-segment spectra that are useful for phonetic discrimination.

It turned out that we obtained better results in the phonetic recognition experiments described below when the complete 145-measurement set was retained instead. However, we did not repeat the GMDA experiments for that set.

Methodology

Hierarchical clustering [Johnson 88] was used to determine sets of similar within-class covariance matrices. In this procedure, distances are first determined between every pair of objects to be clustered. Initially, each cluster consists of a single object. The first step of the algorithm merges the nearest two clusters (objects) into a new cluster. Succeeding steps merge the two nearest clusters according to some criterion for intercluster distance. The process continues until only one cluster exists. Thus, at the i^{th} step of the algorithm, there exist $k - i$ clusters, where k is the number of objects. We denote the distance between the clusters joined on the i^{th} step as H_i, the *height* of the cluster formed on this step. Thus, each of the $k - i$ clusters existing after the i^{th} step have heights less than or equal to H_i. We discuss the procedure for determining distances between covariance matrices below.

We used the *maximum-linkage* clustering method, for which the intercluster distance is defined to be the distance between their most distant members. Empirically, we found that this criterion produced clusters more closely related to manner classes than the average- and minimum-distance criteria that are also commonly used for clustering.

To complete the description of the covariance-clustering method, we must define a measure of distance between covariance matrices. We could have treated the set of elements in each covariance matrix as a vector and used, say, Euclidean distance between the vectors as a measure. However, there is no theoretical justification for such a measure. Instead, we used a measure based on a significance test used to test whether two sample covariance matrices are drawn from populations with equal covariance matrices [Morrison 76]. For the $p \times p$ sample covariance matrices S_1 and S_2 the distance is proportional to

$$[1-\frac{2p^2+3p-1}{6(p+1)}(\frac{1}{n_1}+\frac{1}{n_2}-\frac{1}{n_1+n_2})][(n_1+n_2)\det S_{\text{pooled}}-(n_1 \det S_1+n_2 \det S_2)]$$

where det() is the determinant operator, n_1 and n_2 are the number of data points used to estimate S_1 and S_2, and S_{pooled} is the pooled covariance matrix

$(n_1S_1 + n_2S_2)/(n_1 + n_2)$. The higher the value of this quantity, the less likely it is that the population covariance matrices are equal. Thus, this is a reasonable distance measure. The measure can be extended to test the population equality of more than two covariance matrices.

We used the method just described to cluster the composite segment within-pho covariance matrices for several different measurement sets. The results of the clustering for the 117-measurement set are displayed as a dendrogram in Figure 4.12. Phone labels appear at the left of the dendrogram. Each solid horizontal line in the dendrogram represents a distinct cluster. In particular, the horizontal line extending from each phone label represents the cluster associated with that label's covariance matrix alone. Each solid vertical line represents the merging of two clusters and the position on the vertical axis of the line is the height of the formed cluster. It can be seen that phones with similar properties tend to have similar covariance matrices, according to the distance measure used. For instance, the pair of phones whose matrices are most similar are **ch** and **jh**.

The five clusters with the greatest heights below that denoted by the vertical dotted line were used as groups for the grouped multiple discriminant analysis.. The clusters are demarcated by horizontal dotted lines in the figure. These five groups correspond closely to the vowels, nasals, stops, closures, and fricatives and we will refer to these groups by these names. Note that the vowel group also includes all the semivowels. Phonetic recognition results using the baseline measurement set described in Section 4.2 indicated that the majority of confusions were within these groups rather then across them. Thus, this selection of groups meets the criteria specified above for the potential superiority of GMDA to MDA as a dimensionality reduction method. Consequently, we used these groups in the experiments with GMDA described below. Another good selection would be the two groups formed by cutting the dendrogram at a height between roughly 80000 and 160000. The groups correspond to sonorants and non-sonorants, between which there were very few confusions

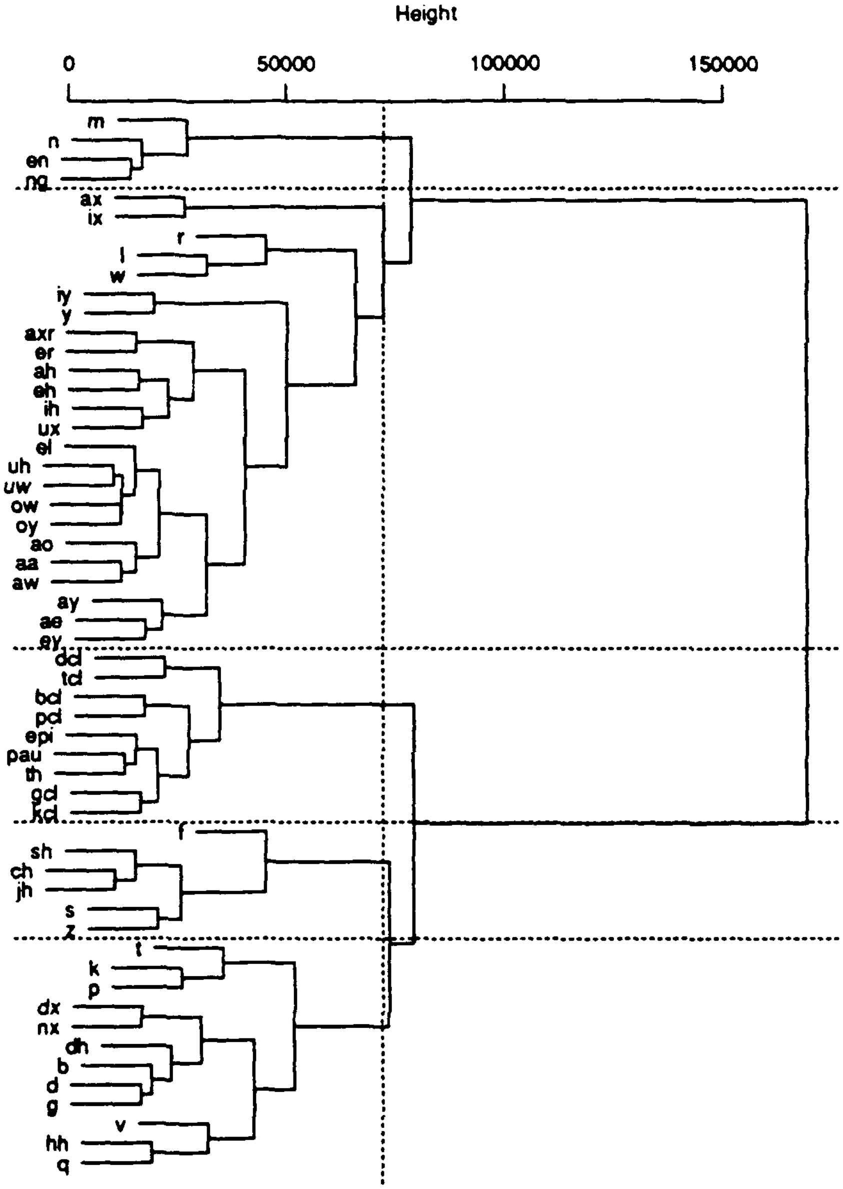


Figure 4.12: Results of composite segment within-phone covariance matrix clustering displayed as dendrogram. Phone labels appear at the left of the dendrogram. Each solid horizontal line in the dendrogram represents a distinct cluster. Each solid vertical line represents the merging of two clusters and the position on the horizontal axis of the line is the height of the formed cluster, in arbitrary units. The five clusters with the largest heights below that represented by the vertical dotted line form the groups used for GMDA, and are demarcated by the horizontal dotted lines.

in the baseline measurement set. However, we did not repeat the experiments for this choice of groups.

One detail: the phone labels **zh** and **em** do not appear in the dendrogram. This is because there were not enough composite segments in the training set associated with any of these phones to obtain a non-singular within-class sample covariance matrix.[2] Thus, a determinant could not be computed for any of these phones' within-class covariance matrices and the phones could not be used in the covariance clustering algorithm. For the purposes of the discriminant analysis, where the small sample size was not a limitation, we arbitrarily assigned these phones to groups that seemed appropriate: **zh** to the fricatives and **em** to the nasals.

Because some confusions in the baseline experiment did occur across groups, we added to the within-group discriminants a set of special-purpose discriminants designed to distinguish phones in particular groups from each other. These were computed by treating groups as classes the purposes of discriminant analysis. Thus, for example, we computed a nasal/vowel discriminant by lumping all the nasals into a class and all the vowels into another class and performing MDA. Note that this generated only a single discriminant because there were only two classes used in the analysis. Likewise, we computed two stop/closure/fricative discriminants and a nasal/closure discriminant. We evaluate the effect of these discriminants on our results in Section 4.8.3.

The total number of discriminants computed by GMDA included fifty within-group- and four between-group discriminants. To form an observation vector of length q from these 54, the set must be ordered by some criterion related to their discrimination power and the first q of these should be included. For MDA, where all discriminants come from the complete set of classes to be discriminated, the best ordering is by the between-class/within-class variance ratios D, as discussed above. However, it is not clear that this criterion

[2] To determine a non-singular sample covariance matrix with a measurement set size of p requires $p + 1$ samples

is appropriate for ordering GMDA discriminants from different groups. For example, assume that the phones in group 1 are less separated in the measurement space than those in group 2. Then a given discriminant y_1 from group 1 might have a lower value of D than a discriminant y_2 from group 2 even though y_1 might be more useful than y_2 for overall phonetic discrimination. Even if each individual discriminant was correctly ordered by its overall discrimination power, it does not follow that the first q discriminants are the best q to use jointly. For example, assume that $q = 2$ and the two discriminants with the highest D values are from different groups. It is possible that the two are highly correlated with each other since they are determined from independent multiple discriminant analyses. Then, once the first of these discriminants is included in the observation vector, the second discriminant might not be the best one to add to the vector. This is not the case in MDA, where all discriminants are determined in a single analysis.

These issues notwithstanding, we used the D values to order GMDA discriminants, subject to the constraint that the number of discriminants selected from each group be roughly proportional to the number computed for that group. In practice, it turned out that we did not have to employ the constraint since the ordering turned out to satisfy it for most values of q.

We used a particular set of these discriminants in the experiments described in Chapters 5 and 6. There were a total of 39 in the set, including the four between-group- and 35 within-group discriminants listed in Table 4.4.

Vowel	Nasal	Stop	Closure	Fricative	Between-group	Total
11	3	9	7	5	4	39

Table 4.4: Distribution of GMDA discriminants for 39-discriminant set.

Interpretation of Vowel Discriminants

To gain some insight into the roles played by the discriminants, we examined the class means of the first few discriminants determined for each group by GMDA. The most interesting results occurred in the vowel group. We present the results in this section.

Figure 4.13 illustrates the class centers plotted on the plane defined by the first two vowel discriminants. The values have been scaled so that the pooled within-class covariance matrix of the discriminants is the identity matrix. Thus, a distance of 1.0 in any direction on the plot corresponds to the pooled within-class standard deviation. The negative of the second discriminant is plotted on the vertical axis so that high vowels (those with typically low values of F_1) appear towards the top and low vowels appear towards the bottom. This suggests that the second discriminant is closely related to F_1. Similarly, the centers are arranged on the first discriminant axis according to their frontness. Front vowels, those with high values of F_2, appear to the left. We have reinforced the impression of the correspondence between the features and discriminants by superimposing a quadrilateral similar to the canonical vowel quadrilateral often used to describe the vowel feature space [Ladefoged 82, p. 75]. We should point out that Pols [Pols 73] obtained a similar result using a slightly different method of discriminant analysis to transform a series of wideband filter energies.

Figure 4.14 illustrates the class centers of the third through the sixth vowel discriminants. The third discriminant apparently corresponds to the feature [+retroflexed] since the /r/-like phones appear by themselves at the right edge of the plot. Also, /l/ appears at the left edge, presumably because it tends to have a high value of F_3. The fourth discriminant class center tends to increase with the amount of spectral movement in the phone so that /ə/ appears at the left edge and diphthongs and /y/ appear at the right edge. It is tempting to associate this discriminant with the feature [+tense] since for the three

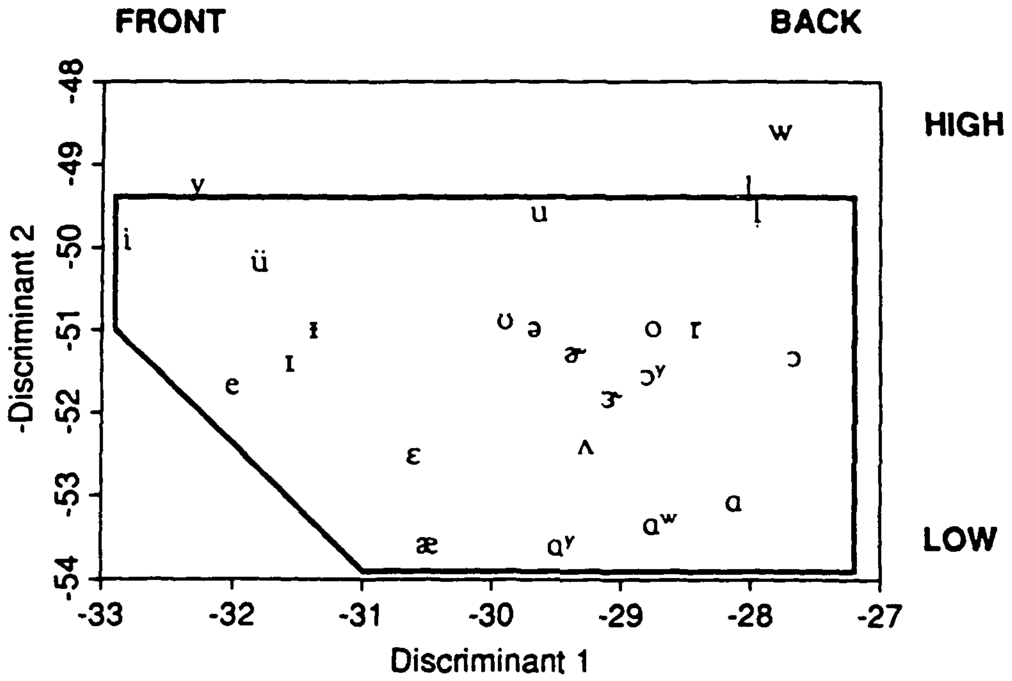


Figure 4.13: Class means of first two vowel discriminants. Classes plotted in IPA. Values scaled so that pooled within-class covariance matrix is the identity matrix. Thus, a distance of 1.0 in any direction on the plot corresponds to the pooled within-class standard deviation. The negative of the second discriminant is plotted on the vertical axis so that the vowels are arranged as in [Ladefoged 82, p. 75]. The quadrilateral is plotted to reinforce the correspondence between the discriminants and feature values.

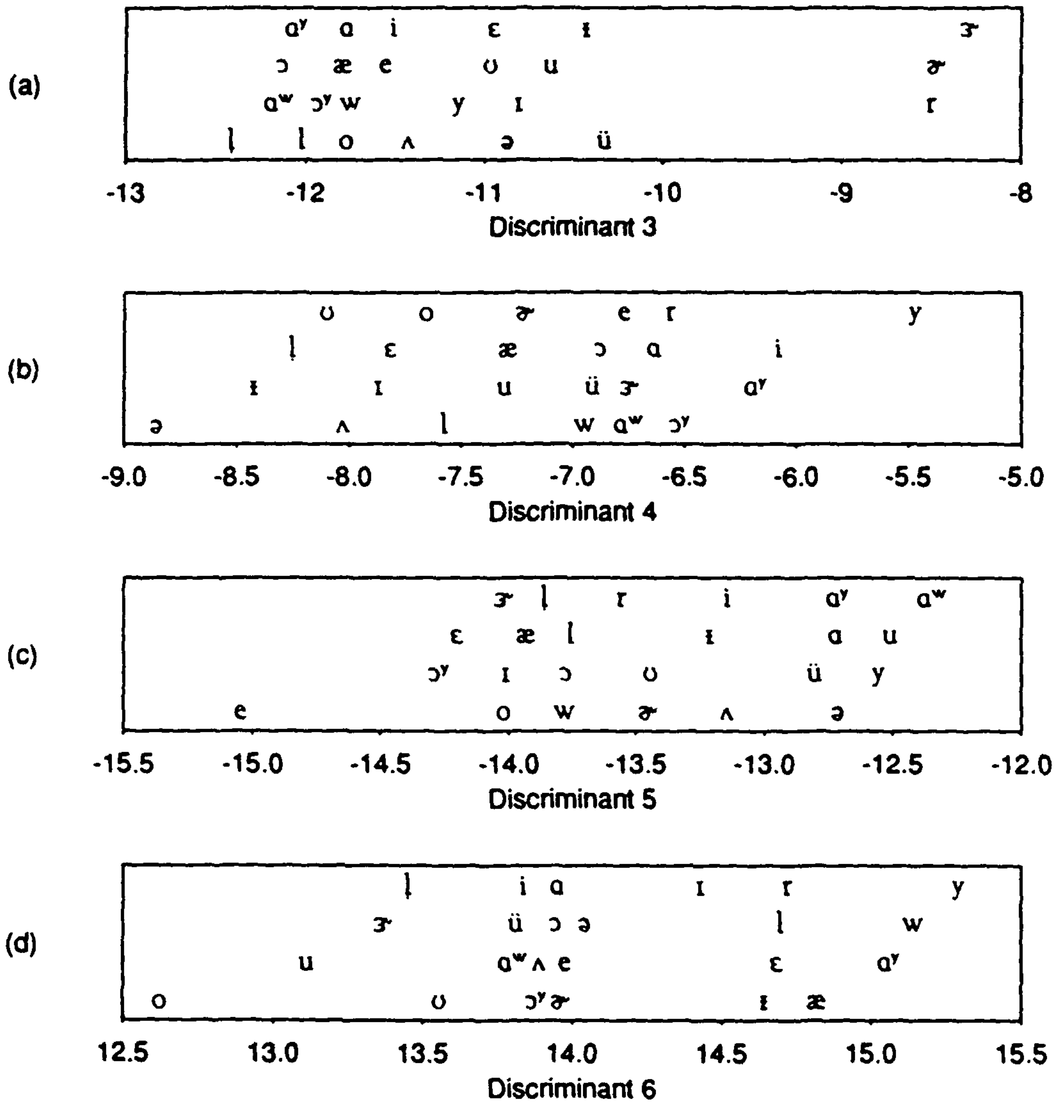


Figure 4.14: Class means of higher-index vowel discriminants. For clarity, labels are placed randomly on vertical axis. Values scaled so that pooled within-class standard deviations are 1.0. (a) Means of the third discriminant. (b) Means of the fourth discriminant. (c) Means of the fifth discriminant. (d) Means of the sixth discriminant.

pairs /ɪ/-/i/, /ɛ/-/e/ and /ʊ/-/u/ that are distinguished only by the value of this feature, the [+tense] vowel center is higher. However, the correspondence does not hold for the /o/-/ɔ/ pair. Higher-index discriminants are harder to interpret. Also, note that class separation decreases with increasing index.

Discussion

Two additional points are worth mentioning about covariance clustering. First of all, the utility of the method for speech recognition is not limited to the application described here. In an HMM-based recognition system that models each state PDF with a multivariate Gaussian, it is expensive to devote specific full covariance matrices for each state in the recognition network. Specifically, if there are N states in the network and the observation vector is of length p, N matrix-vector multiplications, requiring on the order of Np^2 operations, are necessary for each observation vector to compute the observation probability for each state in the network. Additionally, making good estimates of covariance matrices for each state requires lots of training data.

One method for dealing with the estimation problem is to linearly transform the training observation vectors so that their pooled within-state covariance matrix is diagonal [Bocchieri 86]. The same diagonalizing transformation is performed on the observation vectors during recognition. Diagonal covariance matrices can then be used to model each state PDF, reducing the number of operations required per observation vector to the order Np. However, if the within-state covariance matrices differ greatly, even though the pooled within-state covariance matrix is diagonal, this might be a poor assumption for the individual within-state covariance matrices.

An alternative approach would be to cluster the within-state covariance matrices into g groups and to use the pooled within-state covariance matrix for each group to represent the within-state covariance matrices for all states assigned to the group. Each state PDF would be represented by a diagonal covariance matrix. To compute the observation probabilities, the observation vector would undergo g diagonalizing transformations, one for each distinct covariance matrix. Thus, the total number of operations would be of the order $gp^2 + Np$. The system parameter g could be determined based on considerations of training data availability and available computation. Note that $g = N$ corresponds to the state-specific full covariance case and $g = 1$ corresponds to the pooled covariance case described above.

Such a method might also be used in a tied-mixture recognizer [Paul 91, Huang 89, Bellegarda 89] to determine full covariance matrices for some of the mixture components instead of the diagonal covariance matrices typically used. However, details about the specifics of the implementation remain to be worked out.

Another point to raise about covariance clustering is that a technique other than hierarchical clustering could be used and might perform better. The disadvantage of hierarchical clustering using the maximum linkage method is that the within-cluster distance is defined to be the maximum distance between any two of its members. A more useful distance measure would involve all cluster members. In particular, as we stated when we introduced the measure used to define distance between covariance matrices, the measure can be extended to test the equality of any number of sample covariance matrices. Thus, an alternative clustering method would be to predetermine the number of groups g and form clusters so as to minimize the sum of the within-group distances defined by the extension of the significance test. Because of the complicated form of the distance, we do not know if there exists an efficient algorithm for finding a good solution. Developing and testing such an algorithm is beyond the scope of our work and is a subject for further research.

4.8 Phonetic Recognition

We tested the effect of the choice of measurement set on a phonetic recognition task, in which the system attempts to determine the utterance's phone string.

While the usual goal of a speech recognizer is to recognize words rather then phones, phonetic recognition experiments are simpler to run and the results should reflect the relative efficacy of the measurement sets we use. Also, given that in many recognizers, word models are built out of subword models, phonetic recognition performance should be correlated with word recognition performance. We provide evidence for this correlation in Chapter 5, where we compare two of the measurement sets in a word spotting task.

The phonetic recognition task has been used by several researchers, including [Lee 89b, Robinson 91b, Leung 90, Digilakis 92], to test various approaches to speech recognition. While we use a different corpus and slightly different experimental conditions from these researchers, their results are useful benchmarks against which to compare ours.

4.8.1 System Description

To run the phonetic recognizer, the phone and biphone models built as described in Chapter 3 were used to build a network for recognizing phone sequences. The resulting network is illustrated in Figure 4.15. The network can be thought of as a large HMM in which the initial state is connected to the initial state of each phone/biphone model with a null transition, and the final state of each model is connected to both the final state of the network HMM and the initial state of each phone/biphone model. This allows all possible phone sequences. The approach is the same as that used in [Lee 89b].

The Viterbi algorithm [Viterbi 67] was used to determine the highest-scoring state sequence through the network. The hypothesized phone sequence is the sequence of phone/biphone labels of the models associated with the determined state sequence.

As we mentioned in Chapter 3, the training set included 2150 TIMIT utterances from 430 speakers and 473 VOYAGER utterances from 10 speakers. These were used to train the phone/biphone models whose topologies were defined in Chapter 3. The parameters that needed to be trained included the

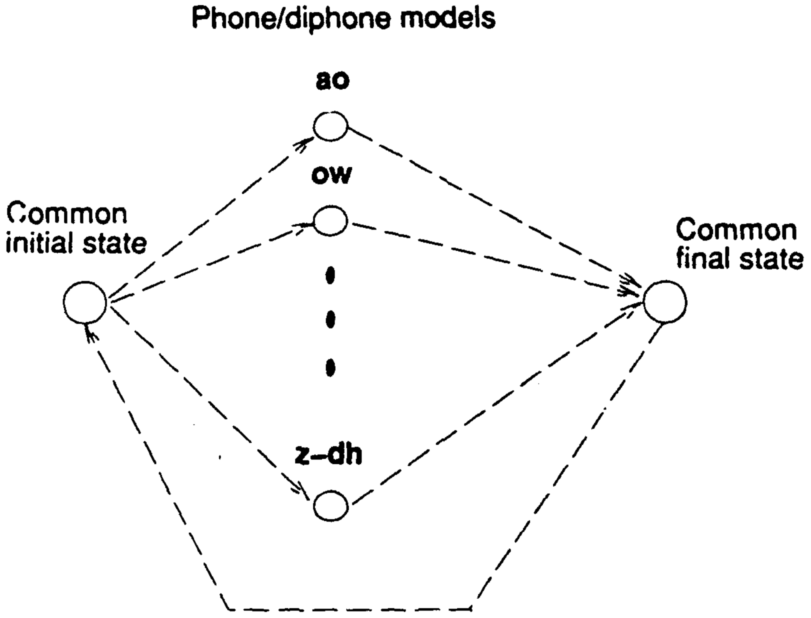


Figure 4.15: Phonetic recognition network. Each circle represents a model such as that described in Chapter 3. Final model states are connected to initial model states. The connection is represented in the figure w:th common initial and final states. However, since a bigram model is used, the arc between the final and initial common states actually represents distinct arcs between every pair of models, each with a distinct transition probability.

within-model transition probabilities, whose training was discussed in Chapter 3, between-model transition probabilities, and state PDF parameters.

As in [Lee 89b], we chose to use a bigram estimate of the between-model probabilities. However. because of the presence of biphone models, a more complex method for estimating the transition probabilities is required. Since the description of our estimation method is fairly detailed and not germane to the main focus of this chapter, it appears in Appendix B.

Each state PDF was modelled as a multivariate Gaussian with a diagonal covariance matrix for the experiments reported this chapter and in Chapter 5. We chose this model because it requires relatively few parameters to estimate compared to mixture or full covariance Gaussian models. This is a particularly important consideration for training biphone models which have few training tokens, and models trained from specific words, which we use in the word spotting experiments of Chapter 5. Also, it requires less computation for training and recognition than either of the other two. On the other hand, it has been shown that increasing the number of mixtures per model in a recognizer can improve performance [Ney 90] so we might have sacrificed performance with this choice.

The state PDF for any state in such a model is represented by an estimated mean vector $\boldsymbol{m}$ and a vector $\boldsymbol{v}$ of weights that are reciprocal of the estimated measurement standard deviations. Thus, each element m_j in $\boldsymbol{m}$ is computed as

$$m_j = \frac{1}{N}\sum_{i=1}^{N} y_{ij}, \qquad 1 \leq j \leq q \tag{4.9}$$

where N is the number of segments associated with the state and y_{ij} is the j^{th} component of the i^{th} training vector and q is the length of the observation vector. Each weight v_j is given by

$$v_j = \sqrt{\frac{1}{\sigma_j^2}} \tag{4.10}$$

where

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(y_{ij} - m_j)^2, \tag{4.11}$$

the sample variance of the j^{th} measurement.

For purposes of both discussion and computation it is convenient to define the score λ that an observation vector is assigned on a state as the log of the state's PDF at the vector. With this definition, the score assigned an observation vector $\boldsymbol{y}$ on a state whose mean and weight vectors are $\boldsymbol{m}$ and $\boldsymbol{v}$ respectively is given by

$$\lambda = -\frac{q}{2}\log 2\pi - \frac{1}{2}\sum_{j=1}^{q}\{[(y_j - m_j)v_j]^2 - \log v_j\} \tag{4.12}$$

which is a special case of the general multivariate Gaussian log PDF [Johnson 88, p. 122] appropriate for the diagonal covariance assumption.

For all measurement sets except the baseline set we applied a linear transformation to the measurements so that the diagonal covariance representation would be more appropriate. We did not do this with the baseline set since we were not particularly concerned with comparing it to the other measurement sets. The transformation diagonalized the pooled within-phone composite segment covariance matrix. We alluded to such a transformation above. Specifically, given the pooled within-phone composite segment covariance matrix $\boldsymbol{S}_{\text{CS}}$ of the measurement set in question, a matrix $\boldsymbol{E}_{\text{CS}}$ whose columns are eigenvectors of $\boldsymbol{S}_{\text{CS}}$ is computed. It can be shown that if each composite segment measurement vector is transformed by multiplying it on the right by $\boldsymbol{E}_{\text{CS}}$ the pooled covariance matrix of the transformed vectors is diagonal. Thus, in the training process, each training segment measurement vector $\boldsymbol{x}^T$ undergoes the transformation

$$\boldsymbol{y}^T = \boldsymbol{x}^T\boldsymbol{E}_{\text{CS}}$$

and the statistics of the transformed vectors y are used in training the models.

We did not have to transform observation vectors determined using conventional MDA because any transformation so determined can be shown to diagonalize the pooled within-class covariance matrix used in the transformation [Johnson 88]. In particular, it can be shown that S_{CS} is diagonal for any set of discriminants determined using MDA. GMDA does not have the same property so we applied the above transformation to the GMDA discriminants.

4.8.2 Evaluation Method

We used a program provided by the National Institute of Standards and Technology (NIST) to evaluate phonetic recognition performance. The same method was used in [Lee 89b] and [Digilakis 92]. Given the utterance's phonetic transcription and the recognizer's hypothesized transcription, the algorithm finds an alignment between them that minimizes the sum of substitutions, deletions and insertions. For example if the actual transcription string is *abce* and the hypothesized one is *bdef* the algorithm finds the following alignment:

$$\begin{array}{ccccc} a & b & c & e & \mathrm{x} \\ \mathrm{x} & b & d & e & f \end{array}$$

where x denotes an insertion or deletion. Thus, in this case, there would be one deletion, one insertion and one substitution for a total of three errors. Note that the match is based completely on the transcription label strings. Label endpoints are not taken into account.

So as to be consistent with [Lee 89b] and [Digilakis 92], we collapsed the complete set of 57 transcription labels into 39 equivalence classes which are shown in Table 4.5. Note that some classes include phonemically distinctive labels so that this scheme will consider confusions between any pair of labels in these classes as being correct. For example, any confusion between a pair of closures will be considered correct rather then a substitution. While we feel this to be a shortcoming of the scheme, it turned out in the few cases we looked at that relative performance among the measurement sets was similar whether the equivalence classes corresponded to the original 57 labels or to the 39 collapsed label sets. Thus, to avoid reporting two sets of results and to

ao aa	m em	s
ow	n en nx	z
aw	ng	sh zh
aa	b	f
ay	d	v
ax ah	g	th
eh	p	dh
uh	t	hh
uw ux	k	ch
ix ih	pcl tcl kcl bcl dcl gcl q epi pau	dx
ey		
ae		
iy	l el	
axr er	r	
oy	w	
	y	

Table 4.5: Confusion classes for phonetic recognition.

be consistent with past work, we will only report results using the collapsed label sets. In general, the difference between the two results is about 10%, (i.e., if 50% of the tokens are correctly recognized in the 39-class scheme, 40% are in the 57-class scheme.)

The test set used for most experiments included 232 VOYAGER utterances from five male speakers. There were 6363 phone tokens. The speaker identifiers are listed in Table 4.6.

ad	jwfm	lmb	reg	sh

Table 4.6: VOYAGER test speakers - Set I.

We used a second test set consisting of four speakers in one experiment where we could not attain a statistically significant result with the five speakers alone, and also for evaluating performance for the best configuration. Experiments involving this set were only run after all measurement sets had been compared for the original five speakers. In particular, the best configuration was determined with the five speakers alone. Thus, the five speakers might be considered a development set and the four a true test set. The four speakers uttered a total of 7224 tokens. Their identities are listed in Table 4.7.

ajd	cph	cth	wch

Table 4.7: VOYAGER test speakers – Set II.

Significance levels and confidence intervals are based on statistics collected on a speaker-by-speaker basis. As pointed out in [Gillick 89], the sample statistics used to test significance or to compute confidence intervals are assumed to be independent evaluations of recognizer performance. However, any two samples drawn from the same speaker are dependent because performance is highly dependent on speaker identity. Thus, we feel it is unjustified to use finer-grain samples, such as the number of errors in each utterance, in the procedures.

4.8.3 Results

Following [Digilakis 92] we will use a figure of merit termed *accuracy* for comparing performance. It is defined as

$$\text{Accuracy} = 100 \times (1.0 - \frac{\#\ \text{substitutions} + \#\ \text{insertions} + \#\ \text{deletions}}{\#\ \text{of actual phone tokens}}.$$

This is equivalent to

$$\text{Accuracy} = \%\ \text{Correct responses - }\%\ \text{Insertions}$$

Middle Frame Measurements

In the first set of experiments, we looked at measurement sets based on the middle frame's spectrum alone to judge the effect of adding the predicted

Measurements	Acc. (%)	Cor. (%)	Ins. (%)	Del. (%)	q
Baseline	44.2	54.2	10.0	12.5	15
MMFSPC	50.0	54.2	4.2	18.8	16
+ MPF$_1$	49.7	54.6	4.9	17.9	17
MMFSPC + MPF$_2$	50.4	55.2	4.8	17.9	17
+ MPF$_1$	50.4	55.5	5.1	16.9	18
+ ENGY	49.2	55.2	6.0	17.2	19

Table 4.8: Phonetic recognition results for middle frame measurements. A measurement set beginning with plus sign (+) includes all measurements of line above as well as measurements after plus sign. Abbreviated columns are, from left to right: (a) Accuracy, (b) Correct, (c) Insertions, (d) Deletions, and (e) Measurement set dimensionality. All measurement sets include duration.

formants and energy measurements. We also compared the MFSC representations to the baseline.

Table 4.8 summarizes the results. While not included in the table, all measurement sets include duration. All results are expressed as a percentage of the number of tokens in the actual transcriptions. The table also includes the number of measurements in each measurement set used. A measurement set beginning with a plus sign (+) includes all measurements of the line above as well as the measurements after the plus sign.

Note that the baseline set performs worse than the others. There are several likely explanations for this. First of all, the state PDF's of the HCEPC's may not be modelled well as Gaussians. This is because the range of hair-cell envelope coefficients is limited to to model auditory threshold and saturation effects [Seneff 88]. Thus, the tails of the state HCE distributions are of finite length, unlike those of Gaussian distributions, which are infinitely long. Since the HCEPC's are linear combinations of HCE's, they are unlikely to be Gaussian. We have informally inspected the state PDF's of several HCEPC's and have found that they indeed deviate substantially from a Gaussian distribution. Also, Glass [Glass 91] has conducted phoneme recognition experiments using Gaussian PDF models and has found that use of mel-based spectral coefficients (MFSC's) yields higher performance than use of HCEPC's. This is the prime motivation behind our use of MFSC's in the other measurement sets investigated. Another reason the baseline performance might be low is that we failed to diagonalize the baseline measurement set as we did for the other sets.

Apparently, adding MPF$_1$ decreases accuracy while adding MPF$_2$ increases it a small amount whether or not MPF$_1$ has been added. We tested the statistical significance of the increase when MPF$_2$ was added by applying a paired one-sided t-test [Lindgren 76, p. 353] to the individual speaker accuracies to test the hypothesis that the mean accuracy was greater when MPF$_2$ was added to the set. When MPF$_2$ was added to the MMFSPC's alone, the significance level was .17 and when it was added to the MMFSPC's and MPF$_1$, the level was .09. Thus, the increase is marginally significant. Finally the addition of energy (ENGY) hurts. There also seems to be a trend towards a higher insertion and lower deletion rate as the number of measurements is increased.

From these results, it appears that the predicted F_1 and energy are not particularly useful measurements for phonetic recognition. Inclusion of the predicted F_2 seems to have a slight positive effect.

As discussed in Section 4.4, it is unclear whether the predicted formants would be more effective if the range over which the models are valid was increased. The resemblance of each formant predictor to a center of gravity computation causes almost all predicted values to lie within the limits of the center of gravity's range. We confirmed this informally by plotting histograms of predicted formants corresponding to true formants of all frequencies. The limited range of the predictor has two ramifications. First of all, the model likely performs poorly in predicting formants outside the ranges, comprising about 8% of F_1 values and 20% of F_2 values. Secondly, because the values fall over a limited range, they may be poorly modelled as Gaussians, just as the HCE's are.

	Acc. (%)	Cor. (%)	Ins. (%)	Del. (%)	q
MMFSPC	50.0	54.2	4.2	18.8	16
AMFSPC	50.1	54.4	4.3	18.2	16
MMFSPC+ $MPF_1 + MPF_2$	50.4	55.5	5.1	16.9	18
AMFSPC+ $APF_1 + APF_2$	49.6	55.2	5.6	16.9	18

Table 4.9: Average vs. middle frame measurements. Abbreviated columns, from left to right: (a) Accuracy. (b) Correct. (c) Insertions. (d) Deletions. (e) Number of measurements in set. All measurement sets include duration.

Middle-Frame vs. Average Spectral Measurements

We next compared the middle frame measurements with those averaged over the segment. As shown in Table 4.9, there was little difference in performance between the two when the two formants were not used but the middle frame measurements did better when the formant measurements were incorporated. This may indicate that the formant measurements are more useful when they pertain to a specific point in time than when they are averaged over the whole segment. However, it is unclear why the same should not be true of the MFSPC's.

Beginning and End Frame Measurements

The next measurement sets tested included beginning and ending frame measurements. As discussed in Section 4.3, these were included to characterize within-segment spectral change. Table 4.10 tabulates results obtained for these sets. Parenthesized expressions obey a "distributive" property so that, for example, the expression $(B + E)(PF_1 + PF_2)$ "expands" into the four measurements BPF_1, BPF_2, EPF_1 and EPF_2 and thus refers to the two predicted formants measured at both the beginning and end frames of the segment.

The results indicate that the addition of the beginning and end frame measurements actually reduces recognition accuracy. The measurement set includ-

Measurements	Acc. (%)	Cor. (%)	Ins. (%)	Del. (%)	q
MMFSPC	50.0	54.2	4.2	18.8	16
+ (B + E)MFSPC	48.9	54.9	6.0	15.6	46
M(MFSPC + $PF_1 + PF_2$)	50.4	55.5	5.1	16.9	18
+ (B + E)MFSPC	49.0	55.7	6.7	14.9	48
+ (B + E)($PF_1 + PF_2$)	47.9	55.2	7.3	14.6	52
+ (B + M + E)ENGY	48.2	55.4	7.3	14.5	55
+ A(MFSPC + ENGY + $PF_1 + PF_2$)	46.7	55.0	8.3	14.6	73

Table 4.10: Effect of begin and end frame measurements. Abbreviated columns from left to right: (a) Accuracy. (b) Correct. (c) Insertions. (d) Deletions. The column labelled q lists the number of measurements in each set. A measurement set beginning with plus sign (+) includes all measurements of row above as well as measurements to right of plus sign. Parenthesized expressions explained in text. All measurement sets include duration.

ing the middle frame MFSPC's and predicted formants exhibits the highest accuracy. In three of the four additions of other measurements to this set, accuracy decreased. The largest set, including all eighteen spectral measurements made at all four within-segment locations (beginning, middle, end and average), performed worst. Once again, the insertion and deletion rates seem to be closely related to the number of measurements.

Dimensionality Reduction

There are two possible explanations for the results tabulated in Table 4.10:

1. The beginning frame, ending frame and average measurements are not useful for phonetic recognition.
2. The measurements are useful but the high dimensionality of the consequent measurement spaces led to poor estimates of model parameters.

To determine whether the high dimensionality was responsible for the poor performance of the larger measurement sets, we applied conventional (not grouped) multiple discriminant analysis to reduce the size of two of the sets in Table 4.10: (a) the 52-measurement set including the MFSPC's measured at beginning, middle and end frames as well as the middle frame predicted formants (described in the fifth row of the table), and (b) the 73-measurement set including all the spectral measurements at all four locations (described in the table's final row). The experiments had two aims: to evaluate the effect of dimensionality reduction on the results and to evaluate the effect of adding the average segment measurements to those of the other three spectra.[3] We used the procedure described in Section 4.7 to perform the discriminant analysis using composite segments.

Figure 4.16 displays the results of these experiments. In the figure, accuracy is plotted against the observation space dimensionality for the two measurement sets as well as for the middle frame measurement set consisting of the MFSPC's and the two predicted formants (the third row of Table 4.10). The latter set was included because it achieved the highest accuracy over all middle-frame measurements. The rightmost points for each of the measurement sets represent the results obtained when there was no reduction of dimensionality. These are the same results as reported in Table 4.10. All other points represent cases in which MDA was used to reduce dimensionality.

The results indicate that reduction of the size of the larger measurement sets leads to a substantial increase in recognition accuracy. However, the 73-measurement set still underperforms the other two by a wide margin regardless of the number of discriminants retained. Thus, adding the average spectral measurements and the energies reduces accuracy. The peak performance of the 52-measurement set (when 18 discriminants are retained) is just slightly

[3] We inadvertently used the 52-measurement set in these experiments rather then the 55-measurement set described on the sixth row of the table. Thus, the larger measurement set differs from the smaller one in that it includes the energy measurements made at the beginning, middle and end segment positions as well as the average measurements. However, the small difference in performance observed between the 52-measurement and 55-measurement sets suggest that the results reported here would not be substantially different if the 55-measurement set were used.

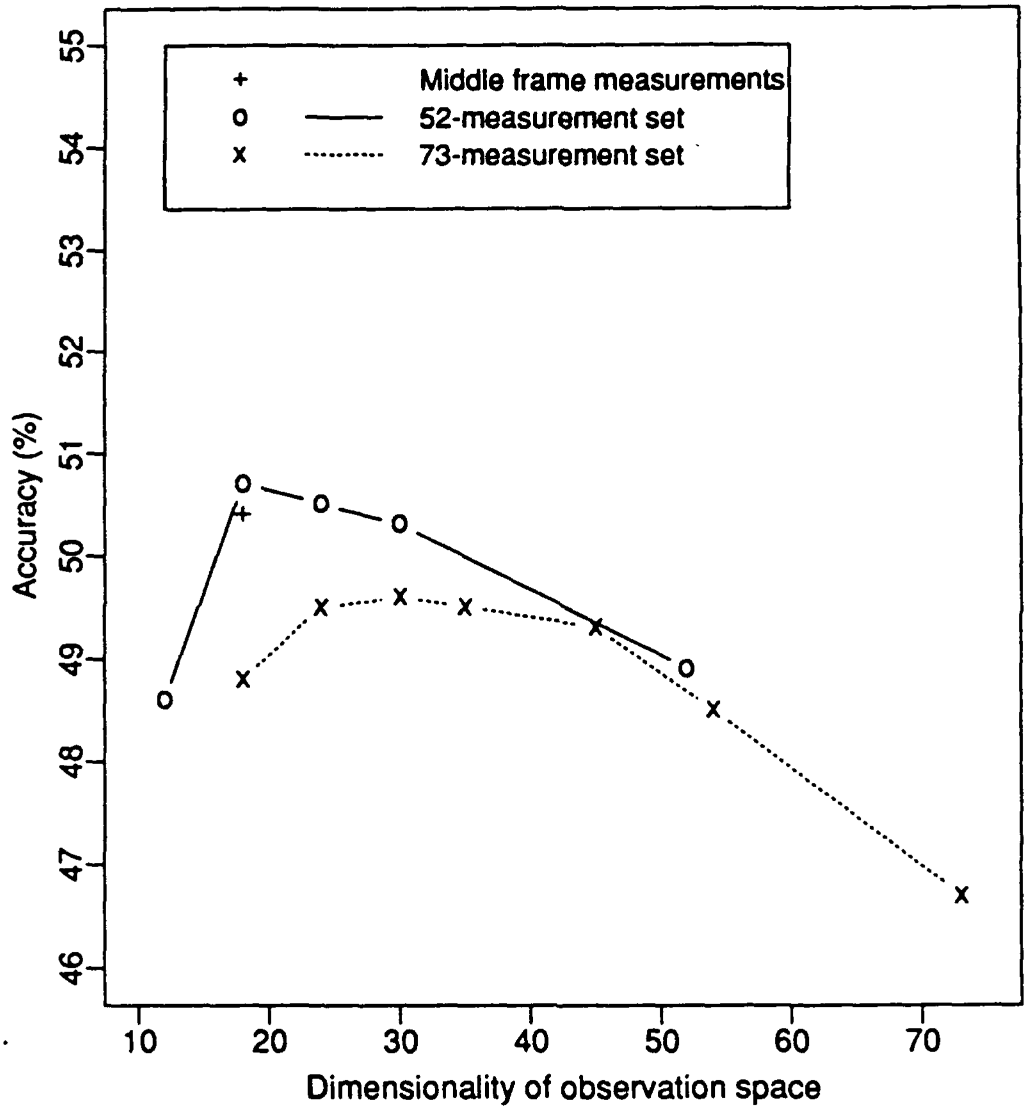


Figure 4.16: Phonetic recognition as a function of dimensionality of observation space. Accuracy is plotted as a function of the observation space dimensionality for the measurement set including the middle frame MFSPC's and predicted formants, and the 52- and 73-measurement sets described in text. The rightmost point for each measurement set represents the results for the unreduced set. All other points represent results obtained through dimensionality reduction through conventional multiple discriminant analysis.

better than that of the middle-frame measurement set, suggesting that there is not much gain from adding the begin and end frame measurements.

In light of the fact that the inclusion of *measurements* of spectral change in frame-based systems is known to improve performance, e.g., [Furui 86, Lee 88], this result might be surprising. However, there is a plausible explanation. Within-region measurements of spectral change are probably most important for characterizing phones whose regions tend to exhibit the most change (e.g., diphthongs). If each token of such a phone had only one segment associated with it, within-segment measurements of spectral change would thus be important. However, in our system, tokens of such phones tend to have several segments associated with them, since, as shown in Chapter 3, the number of segments associated with each phone increases with the magnitude of the phone's within-region spectral change. Furthermore, the HMM topology we employ usually forces each segment associated with a given token to match a distinct state of the HMM. Thus, for any such phone, within-region spectral change is likely characterized by differences in parameters among states in the phone's model. Thus, for phones associated with large spectral change, adding measurements at several locations within the segment may be redundant. For other phones, there is likely little benefit from these measurements either.

If this hypothesis is correct, then there is an interaction between the segmenter and the measurement set. For instance, the ideal segmenter described in Chapter 3, that produces one segment per phone regardless of phone type, might benefit to a greater degree than ours from measurements that capture within-segment spectral dynamics. This would hold true as well for the stochastic segment model [Ostendorf 89, Zue 89a, Digilakis 92]. Thus, further experiments to test this hypothesis would likely be worthwhile.

Out-of-Segment Measurements

The final set of measurements considered in the phonetic recognition experiments consisted of the out-of-segment measurements made at 5 and 35 ms beyond the segment edges. For this set of experiments, all 145 measurements made at all positions summarized in Section 4.6 were included. Thus, the set included the MFSPC's, energies and predicted formants made at begin. middle and end frame positions, the within-segment averages of these measurements, and the out-of-segment measurements. We term this the complete measurement set. Multiple discriminant analysis was used to reduce dimensionality.

A plot of accuracy as a function of the number of discriminants retained in the complete measurement set appears in Figure 4.17. For purposes of comparison, the figure includes a similar plot for the 73-measurement set described above as well since the only difference between the two sets is the inclusion of the out-of-segment measurements. Because there were 55 classes included in the discriminant analysis, the maximum number of available discriminants was 54. It is clear from the plot that the inclusion of out-of-segment measurements provides a substantial improvement in performance regardless of the number of discriminants used.

We tested the statistical significance of this result by applying the paired one-sided t-test to the individual speaker accuracies to test the hypothesis that the mean accuracy was greater for the complete set than for the 73-measurement set. We compared the results for the 30-discriminant case for both measurement sets. This case was chosen because it led to the best performance on the 73-measurement set and we wanted to be conservative in testing the hypothesis that the out-of-segment measurements improved performance. The difference in means was significant with $p = .014$. Thus, including the out-of-segment appears to be advantageous. The results by speaker are listed in Table 4.11.

Because the positions of the out-of-segment spectra were selected so as to maximize stop discrimination, we expected that the reduction in confusions among stops would be mainly responsible for the improved results. To check this, we constructed confusion matrices for the 73-measurement and complete measurement sets (both reduced to 30 discriminants), for which the confusion

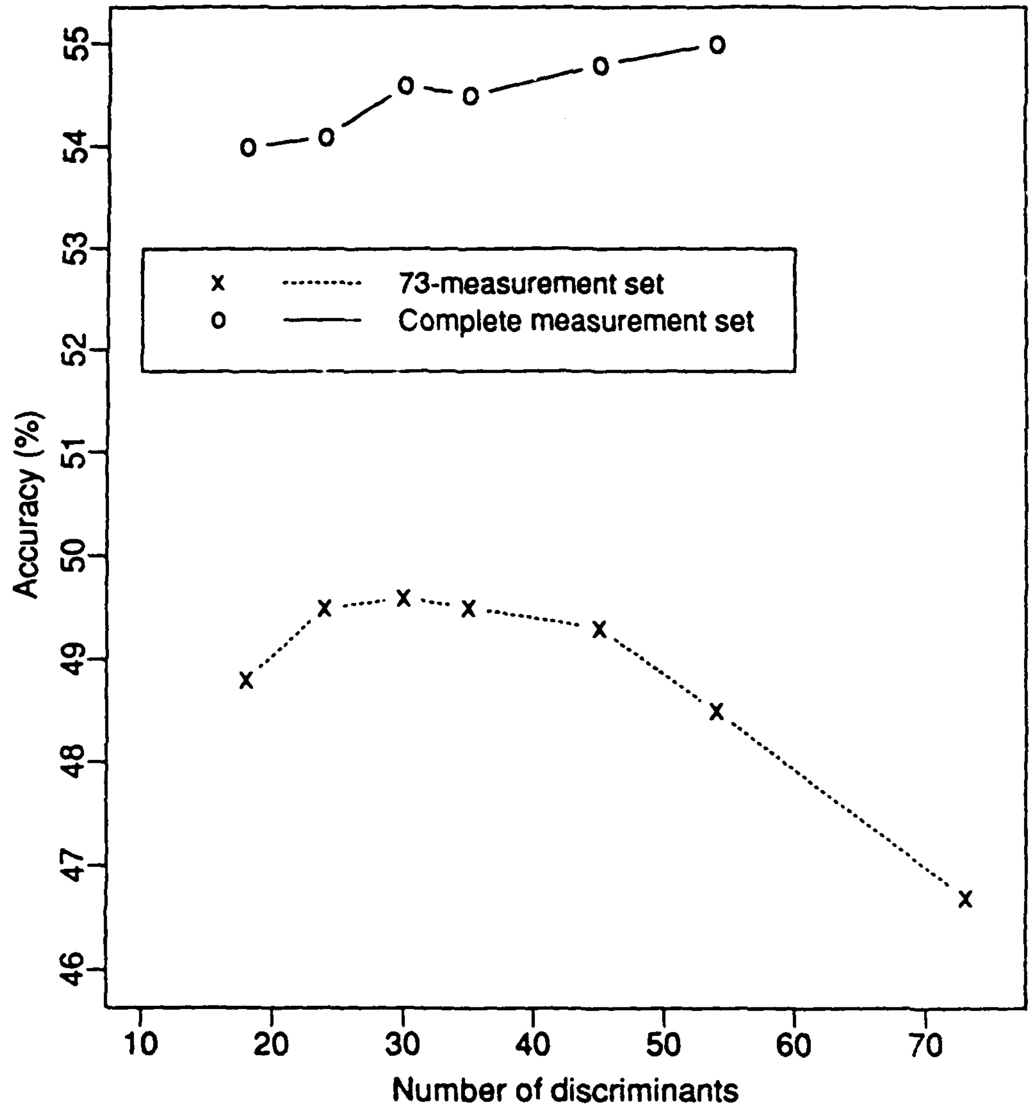


Figure 4.17: Effect of out-of-segment measurements on phonetic recognition. Accuracy is plotted as a function of the number of discriminants retained in the discriminant analysis for the two measurement sets, which are described in the text. The result obtained for the 73-measurement set when there was no dimensionality reduction is plotted as well for consistency with Figure 4.16.

	ad	jwim	lmb	reg	sh
73-meas	47	41	53	55	53
Complete	54	40	63	60	58

Table 4.11: Measurement set comparison by speaker. The compared sets are the 73-measurement and complete measurement sets, each reduced to 30 discriminants. Numbers are recognition accuracies.

classes were vowels, semivowels, fricatives, nasals, stops and closures. We then computed the differences in numbers of errors between the two cases. The results are tabulated Table 4.12.

The tables express the differences in numbers of errors in two ways. Each cell in the top table displays the percentage of the total reduction in errors accounted for by the confusion type associated with that cell. Thus, a large value indicates that the reduction in that type of error played a big role in improving overall performance. For example, the largest error reduction was in the number of times that a vowel was incorrectly inserted in the hypothesized transcription since the reduction in this type of error accounted for 14% of all the improvement in performance, more than for any other reduction. Each cell in the bottom table expresses the percentage reduction in the type of error. For example, the number of confusions for which the input class was a stop and the output class was a nasal was reduced by 76% through the introduction of the out-of-segment measurements. Because there were far fewer of these confusions than there were vowel insertions, this only accounted for 6% of the total error reduction.

The reduction in errors is widely distributed among most types of confusions. However, the number of stop-stop confusions hardly dropped at all even though the positions for making the out-of-segment measurements were chosen to maximize the separation in the measurement space among stops. In fact, this type of error is one of the most common regardless of the measurement set used (see Table 4.15). Thus, phonetic recognition would benefit from more

		Proportion of total error reduction(%)							
		Output class							
		Vow	Semi	Fric	Nas	Stop	Clos	Del	Total
Input class	Vow	8	3	4	-1	-1	3	-9	6
	Semi	1	-1	1	2	0	0	2	6
	Fric	1	3	5	3	3	1	-4	10
	Nas	5	0	0	4	1	0	0	11
	Stop	2	1	4	6	1	-2	7	19
	Clos	4	2	9	3	2	NA	2	23
	Ins	14	0	7	2	4	-1	NA	25
	Total	35	8	30	18	9	0	-2	100

		Reduction in number of errors by type(%)							
		Output class							
		Vow	Semi	Fric	Nas	Stop	Clos	Del	Total
Input class	Vow	5	17	18	-4	-18	38	-9	2
	Semi	5	-20	13	26	17	0	6	7
	Fric	7	62	14	33	20	10	-9	9
	Nas	37	13	11	22	18	0	1	14
	Stop	9	19	23	76	3	-30	13	13
	Clos	19	50	31	30	21	NA	7	21
	Ins	28	0	28	10	17	-10	NA	18
	Total	12	18	21	22	10	1	-1	10

Table 4.12: Phonetic recognition error reduction due to inclusion of out-of-segment measurements, by confusion type. The two cases compared are the 73-measurement and complete measurement sets, each reduced to 30 discriminants. The top part displays the proportion of the total error reduction for each confusion type, expressed as a percentage. There were 318 fewer errors for the complete set. Thus, for example, since there were 25 fewer vowel-vowel confusions for the complete measurement set, the proportion was 25/318 = 8% of the total error reduction. The bottom part displays the percentage reduction in number of errors for that type of confusion. For example, there were 495 vowel-vowel errors when the 73-measurement set was used and 470 when the complete set was used, corresponding to a reduction of 5%. Negative numbers indicate an increase in number of errors. Classes in order are vowel, semivowel, fricative, nasal, stop, and closure. Deletions and insertions tabulated as well. The "insertions" row tabulates the inserted labels by class.

work on the problem.

The Effect of Biphone Models

As discussed in Chapter 3, a key difference between our system and a conventional frame-based HMM recognizer is that ours requires biphone models. We pointed out that a potential drawback of their use is the relative scarcity of data for training them. To investigate the effect of biphone models on the phonetic recognition results, we identified the sequence of models hypothesized by the recognizer for each utterance, identified the biphone models in the sequence, and computed the recognition rate for all phone tokens aligned to these models according to the NIST evaluation algorithm. For instance, where a model sequence includes the **aa-r** model, the phone sequence /ɑr/ is hypothesized. If the evaluation algorithm aligns these hypothesized phones to the sequence /or/ then one out of the two tokens aligned to the biphone model is correctly recognized. We repeated this procedure for the phone models in each model sequence and compared the results. Note that these statistics only pertain to insertion and substitution errors, since deleted phones, by definition, are not aligned to any model.

	Correct(%)	Insertions (%)	No. hypothesized phones
Phone models	69	5	10300
Biphone models	72	10	2368

Table 4.13: Biphone model performance. Percentages computed over number of hypothesized labels for nine speakers in both sets.

Table 4.13 summarizes the results of this investigation for the complete measurement set reduced to 54 discriminants. This configuration was chosen because it performed the best of all investigated. The results pertain to the nine speakers in both test sets. Note that in contrast to the results we have reported previously, the percentages in the table are based on the number of hypothesized labels rather then the number of actual transcription labels. The results indicate that phones aligned to biphone models are actually *more* likely to be correct than those aligned to phone models.

While this effect is offset by the higher insertion rate for these models, the result is still somewhat surprising. It appears that the potential advantages of biphone models cited in Section 3.5 might outweigh the disadvantage of training sparsity. Given the number of arbitrary decisions we made in selecting the biphone model inventory and training sets, their performance can probably be made even better.

Another point to note is that biphone models account for about 19% of all hypothesized phones while the statistics of the segmenter on the training set that were tabulated in Table 3.5 imply that less than 13% of phones are merged into biphones. Thus, there seems to be a bias towards these models. It is not clear why this is so. However, the large number of appearances of the biphone models in the alignments indicates that their observed performance is a reliable indicator of their ability to model the test data.

Comparison of MDA and GMDA

For the comparison of multiple and grouped multiple discriminant analysis, we used both techniques to reduce the dimensionality of the 117-measurement set described in Section 4.7.3. Our choice of this measurement set for the comparison was somewhat arbitrary and was made before performing the phonetic recognition experiments described above. The comparison could as easily have been made for the 145-measurement set that yielded the best recognition results.

We also compared two sets of GMDA discriminants, one which included the four between-group discriminants described in Section 4.7.3 and one which did not. The results for the three sets of discriminants are displayed in Figure 4.18.

The grouped multiple discriminant analysis results are slightly better than those for conventional multiple discriminant analysis over most of the range

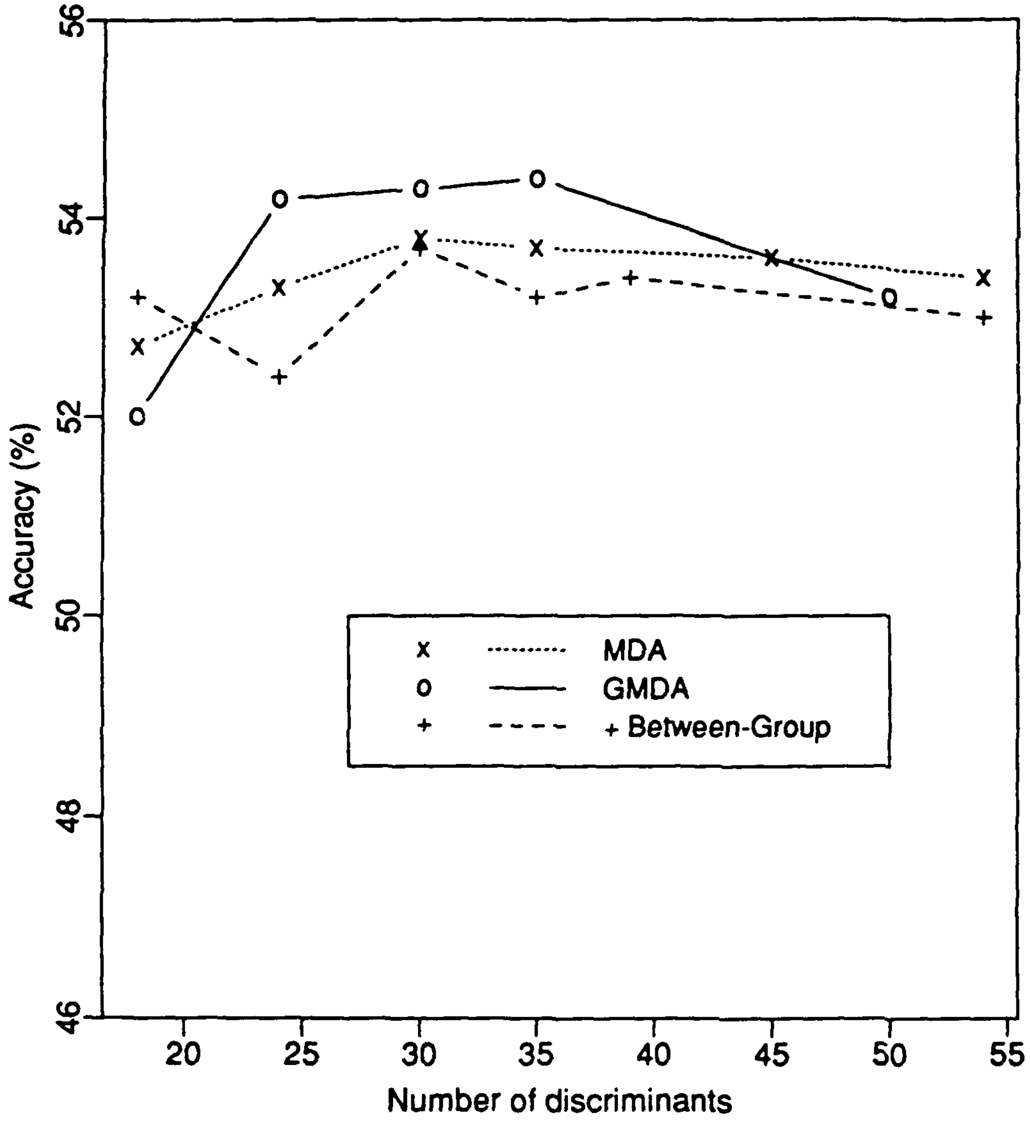


Figure 4.18: Comparison of MDA and GMDA. Accuracy is plotted as a function of the number of discriminants retained in the discriminant analysis. The results labelled + between-group are for observation vectors that include the four between-group discriminants defined in Section 4.7.3.

in number of discriminants retained. We chose the 30-discriminant case to test whether GMDA is significantly better than MDA and obtained a level of $p = .35$. As this did not indicate significance, we redid the comparison after the four speakers in the second test set were included. For the nine speakers, the GMDA results were actually slightly worse. Thus, while GMDA produces comparable results, it does not appear to be an improvement over MDA.

One point to note on the plot is that there is a large increase in GMDA's performance when 24 discriminants, rather then 18, are used. This suggests that discriminants between the 18th and 24th are important for discrimination. In fact, of all three schemes in which 18 discriminants are used, GMDA performs the worst. It is therefore likely that the GMDA discriminants are not ordered properly (i.e., some of those between 18 and 24 should have been included in the first 18). As discussed in Section 4.7.3, there is some question of how to order discriminants produced by GMDA and so this result is not surprising.

We have not compared the methods on the other measurement sets investigated in this chapter. In particular, we have not compared them on the complete measurement set, which exhibited the highest accuracy over all sets investigated. Doing so would be worthwhile for evaluating the generality of the results. Also, as we pointed out in Section 4.7.3, it might be worthwhile to repeat the experiment using the sonorant and non-sonorant groups. Finally, as we pointed out in the description of the technique, it might be worthwhile to experiment with alternative methods for clustering covariance matrices and for ordering the discriminants. Perhaps the GMDA technique could be shown to outperform MDA in these cases.

The Best Results: Summary and Comparison to Previous Work

The highest accuracy on the first test set was 55.0% and was achieved with the complete measurement set reduced to 54 discriminants. For the second test set, it was 63.2%. For the remainder of the section, we will report results

Speaker	Acc. (%)	Cor. (%)	Ins. (%)	Del. (%)	N
jwfm	42	47	5	25	1427
ad	53	59	6	14	1154
ajd	58	64	6	11	1951
sh	58	63	5	16	1260
reg	61	68	8	9	1276
lmb	62	67	4	10	1246
cph	63	70	7	8	1737
wch	66	69	3	12	1780
cth	66	71	5	9	1756
Total	59	65	5	12	13589

Table 4.14: Results by speaker on best-performing measurement set, arranged in ascending order of accuracy. Abbreviated columns denote accuracy, correct, insertion and deletion rates. N is number of actual phone labels.

for the joint set of nine speakers. Table 4.14 summarizes the per-speaker and overall statistics for this set. The mean speaker accuracy was 58.9%. The 95% confidence interval for the mean is 53.1-64.7%, assuming a t-distribution for the sample speaker accuracies.

Note that performance for speaker "jwfm" is comparatively very poor. In particular, there was a very high deletion rate for this speaker. From Table 4.11 it can also be seen that for this speaker alone, use of the complete measurement set did not lead to any improvement over the 73-measurement set. We have not investigated the reasons for these differences. The wide variation in performance among speakers is consistent with results reported in the literature for isolated and continuous speech recognition tasks and suggests that to achieve high recognition accuracies, future work should focus on understanding and dealing with interspeaker variability. The number of errors by confusion class, expressed as percentages of total errors and of the number of input tokens of each class is tabulated in Table 4.15.

As discussed above, Lee and Hon [Lee 89b], Leung et al. [Leung 90], Robinson [Robinson 91b] and Digilakis [Digilakis 92], have used the phonetic recog-

		Proportion of total errors (%)							
		Output class							
		Vow	Semi	Fric	Nas	Stop	Clos	Del	Total
	Vow	16	2	2	2	1	1	11	34
	Semi	3	1	1	0	0	0	4	9
Input	Fric	1	0	5	1	1	1	4	12
class	Nas	1	0	1	2	1	1	3	7
	Stop	2	1	2	0	5	1	4	15
	Clos	2	0	3	0	1	0	4	9
	Ins	4	1	2	2	2	2	NA	13
	Total	29	5	14	7	11	5	30	100

		Error rate (%)								
		Output class								
		Vow	Semi	Fric	Nas	Stop	Clos	Del	Total	N
	Vow	19	2	2	2	1	1	13	39	4833 (36%)
	Semi	15	3	3	2	2	0	17	43	1199 (9%)
Input	Fric	3	1	12	2	2	2	10	32	2165 (16%)
class	Nas	2	1	3	8	2	3	13	32	1176 (9%)
	Stop	5	1	5	1	15	2	12	40	2021 (15%)
	Clos	4	0	7	1	2	0	10	24	2193 (16%)
	Ins	31	9	14	12	19	14	NA	100	740 (6%)

Table 4.15: Confusions by class for best performing measurement set. Each cell in top table is the proportion of total errors due to the confusion type associated with that cell, expressed as a percentage. Each cell in bottom table is ratio of the number of errors of that confusion type to the number of input tokens of the given class. For example, the numbers in the vowel-vowel cells were computed as follows: there were a total of 5520 errors, of which 904, or 16% involved vowel-vowel confusions. These 904 comprised 19% of the 4833 vowels in the test set. Classes in order are vowel, semivowel, fricative, nasal, stop, and closure. Deletions and insertions tabulated as well. The "insertions" row tabulates the inserted labels by class. *N* is total number of labels of each class. Percentages in right column denote proportion of labels of each type.

nition task to evaluate approaches to speech recognition. Before discussing their results, we should point out differences between these experiments and ours which make a direct comparison impossible.

First of all, each of these researchers used the TIMIT corpus to test their systems while we used the VOYAGER corpus. We are unsure of the effect of this difference. Secondly, they used speakers of both genders in testing and training while we used male speakers only, thus making our task easier. As mentioned in Chapter 2, we also removed silences and non-speech events before and after each utterance based on the phonetic transcription rather then model them explicitly. This probably had a small negative effect on our results, based on a performance comparison cited in [Robinson 91b] between label inventories in which silence is and is not retained. Presumably, the recognition rate of the beginning and ending silences is higher than that of the other labels. A final point to consider is that previous researchers used the "sx" and "sa" utterances in the TIMIT corpus for both training and testing. Since the "sx" set includes only about 450 distinct orthographies and since there were well over 450 "sx" sentences used in training for each of the experiments , there was likely considerable overlap between training and testing sentences. Thus, training was vocabulary-dependent, an effect known to have a positive effect on performance [Hon 89, Hetherington 91]. Such was not the case in our work, since 80% of our training data was from the TIMIT corpus while the test data was drawn from the VOYAGER corpus.

These provisos notwithstanding, making the comparison is useful since it can provide a rough idea about whether the segment-based HMM's performance is comparable to that of other approaches. In fact, we do not believe that the cumulative effect of the differences between our experiments and previously reported ones is large enough to change our ultimate conclusion: namely, that the segment-based HMM recognition performance is competitive with that of existing systems of similar computational complexity.

Table 4.16 summarizes results of previous experiments and how the condi-

Reference	Acc. (%)	Complex.	Conditions
Current work	59	4	Described above.
[Lee 89]	53	2	Frame-based HMM with 48 context-independent models. Glottal stop labels removed for evaluation.
	66	NA	Same but with 1450 context-dependent models.
[Leung 90]	47	?	Baseline SUMMIT system (no bigram). 38 labels in evaluation.
	50	7	Same but with connectionist classifier.
	56	200	Same but with less constrained segment network.
[Robinson 91a]	75	11	Recurrent neural network. Label inventory as in [Lee 89].
[Digilakis 92]	66	80	Stochastic segment model.

Table 4.16: Previously reported phonetic recognition results. Abbreviated columns denote accuracy and complexity. The complexity is an estimate of the number of multiplications (in millions) required in the segmentation and recognition of a three-second utterance. The complexity estimates are outlined in text. Those with question marks were not estimated. For the context-dependent model, the computational requirement is probably dominated by the Viterbi alignment calculation and so the complexity estimate is not a good measure of it. Except for our work, all experiments used the TIMIT corpus for testing. Unless otherwise noted, a bigram phonotactic model and context-independent acoustic models were used.

tions differ from ours. We discuss these results in more detail below. Unless otherwise stated, a bigram phonotactic model and context-independent acoustic models were used in each experiment.

As can be seen from the table, Robinson [Robinson 91b] reports the highest accuracy, 75%. This represents the state of the art. With the use of 1450 context-dependent models, Lee and Hon [Lee 89b] obtained an accuracy of 66% (74% correct and 8% insertions). Their results with context-independent models are more relevant for comparing to our work since our models did not take context into account.[4] With 48 context-independent models, the accuracy

[4] It can be argued that biphone models take context into account. However, as discussed

was 53% (64% correct and 11% insertions), also with a bigram model.

The results achieved using context-independent models in [Digilakis 92] are better than those reported by us or by Lee and Hon. However, as shown in the table, this system appears to require much more computation, if one considers the number of multiplications required per utterance for segmentation and for segment scoring. This is usually a reasonable estimate of computational complexity for phonetic recognizers since most of the computation is devoted to these tasks. However, if many models are used, as in the context-dependent model of in Lee and Hon's work, the Viterbi alignment probably dominates the computation. Thus, we have not attempted to estimate the complexity for that case. The "Complexity" column of the table specifies the number of multiplications in millions for a three-second utterance.

For instance, the system of Lee and Hon uses three vector quantizers to compute frame scores. In vector quantization, a Euclidean distance must be computed between a measurement vector and each cluster center. For a measurement vector with q elements, each Euclidean distance calculation involves q multiplications. Thus, for a VQ codebook of size K and a vector of size q, Kq multiplications are required per quantizer. For all three quantizers, $K = 256$ and the three vectors used are of size 12,12, and 2. The quantization must be done once per frame. For the reported frame rate of 100/second, a 3-second utterance would require about 2 million multiplications.

By comparison, the best results obtained by Leung et al. involved a neural network scorer with 30,000 connections. For the best results reported (56% accuracy), an average of 7,200 segments had to be scored for the average three-second utterance. The number of multiplications for an utterance is thus roughly $7,200 \times 30,000$ or about 200 million. When a more constrained search involving 240 segments is used, this number is about 7 million but accuracy drops to 50%.

above, these only matched 17% of all input phone labels. Additionally, the model parameters were not smoothed with context-independent estimates, as is usually done when context-dependent models are used.

The system in [Robinson 91b] requires about 11 million multiplications, according to [Robinson 91a], which describes essentially the same network as that used in [Robinson 91b].

A corresponding number is difficult to ascertain from Digilakis' work but he does report an average of 60,000 "Gaussian computations" per three-second utterance when using a bigram model and an efficient search algorithm.. Each Gaussian computation for the block-diagonal "independent frame" model involves a multiplication of a full $q \times q$ covariance matrix by a vector of length q where q is the size of the observation vector. Thus, each involves about q^2 operations. For the reported results, $q = 37$ and the number of computations is about 80 million.

In our system, the segmentation step requires a multi-level segmentation to be computed and each segment above the seed regions in the MLS to be assigned a probability of being a merge, split or good segment. The MLS computation itself requires Euclidean distances computed between each adjacent pair of 5 ms frames to generate seed regions. If, as is the case here, 40 hair-cell coefficients are used to represent each frame, this requires $200 \times 3 \times 40 = 24000$ computations. An even smaller number is required to generate the MLS once the seed regions have been computed. Thus, the total MLS computation is negligible.

Each merge/split/good probability assignment is performed using vector quantization and thus requires a Euclidean distance computation. Our segmenter uses a codebook of size 256 and characterizes each segment with 21 measurements. For a typical three-second utterance, probabilities are computed for 120 segments. This is the average of the number of segments in the multi-level segmentation, not including seed regions. Thus, the segmenter requires $256 \times 21 \times 120 \approx 0.6$ million multiplications.

For phonetic recognition, a weighted Euclidean distance must be computed for each acoustic segment for each state in each model. This computation requires twice the number of computations as a Euclidean since the weights must be multiplied by the elementwise distances. Thus, if the number of states is S, the number of segments per utterance is T, and the observation vector is of size q, $2STq$ operations are required. In our system, $S = 464$, $T \approx 60$ for a three-second utterance and $q = 54$. Thus about 3 million multiplications are required for phonetic recognition and 4 million altogether for both segmentation and phonetic recognition. We should point out that accuracy on the five-speaker test set when 18 discriminants were used was within 1% of those obtained for our best system, when 54 discriminants were used. The recognizer in this case would require a total of 1.5 million multiplications for a three-second utterance, making the system the least costly of those compared.

We have made this comparison to show that recognizers that employ more complex PDF models (such as full-covariance Gaussians) or segmenters tend to achieve better performance. Thus, for example, were we to replace our diagonal covariance Gaussian PDF assumption by a more complex model such as a mixture or full covariance Gaussian, our computation requirements would increase but presumably our performance would as well, since those models are more general than diagonal Gaussians. We discuss other possible improvements to the system in the next section.

4.9 Summary and Discussion

This chapter consisted of three parts: an investigation of how acoustic-phonetic knowledge can be represented in the measurements made on acoustic segments, a comparison of phonetic recognition performance for different measurement sets, and a comparison of the phonetic recognition performance of the segment-based HMM to that of existing approaches to speech recognition.

We performed several experiments to investigate knowledge representation. Each of these experiments had two aims: to investigate, for its own sake, how knowledge is represented in the measurements, and to use the results of these investigations to improve phonetic recognition performance. We were quite successful in developing insight into knowledge representation but less so in applyii.g the results to phonetic recognition.

For example, we showed that formants, which play a large rcle in theories of speech perception and production, are not modelled well as a linear combination of mel-frequency spectral coefficients. However, we were able to build good models for the first two formants that were valid for most formant values observed in the data by making nonlinear transformations of the coefficients. Thus, we showed that the representation of formants by the MFSC's is nonlinear. The nonlinearity suggested that there might be a benefit to adding the modelled formants to the measurement set. The potential benefit is due to the fact that decision boundaries computed by the diagonal covariance Gaussian probability are linear. If phonetic discrimination is strongly related to differences in formant frequencies among different classes then adding the formants directly to the measurement would tend to make the decision boundaries more linear in the space used by the classifier. Thus, the classifier could have a more accurate model of the decision boundaries and could potentially perform better. However, only a small improvement was noted when an estimate of F_2 at the middle frame of the segment was added to other middle-frame measurements. Adding an estimate of F_1 actually reduced performance somewhat.

In another experiment, we chose positions fcr making spectral measurements in segments adjacent to the one being modelled by optimizing a measure of class separation for voiced stops. While the results of the optimization experiment suggested that spectral measurements made at these points would reduce the number of confusions among stops, the inclusion of these measurements in the phonetic recognizer did not do so. However, it led to a large and statistically significant improvement in overall performance. Thus, in that experiment, we were successful in applying the knowledge that measurements made outside segments are important for phone discrimination but were unsuccessful in focusing this knowledge to reduce the number of a particular type of confusion.

Finally, we showed that, as predicted, within-class covariances vary widely by phone label but clustered well by manner of articulation. This finding led us to develop an alternative to multiple discriminant analysis that attempts to overcome its faulty assumption of equal class covariance matrices. The new method determines a distinct set of discriminants for each group determined by clustering covariance matrices. We showed that for the group of semivowels and vowels, several of the discriminant functions were closely related to distinctive features, implying that that certain distinctive features are well-represented linearly in the set of mel-frequency spectral coefficients. However, we did not observe an improvement in phonetic recognition performance when using the new method instead of conventional multiple discriminant analysis to reduce dimensionality.

Thus, we conclude that it is difficult to predict how recognizer performance will change when one attempts to improve knowledge representation. The prediction is difficult mainly because there are many processes that intervene between the steps taken to improve knowledge representation and the recognizer's final output. The processes include the loss of information due to dimensionality reduction, mismatches between the actual and assumed probability distributions for the measurements, and the segmentation process, to name just three. We believe that to reap the benefits of better knowledge representation, it is important to analyze recognizer behavior in detail, focusing on the effect of these invervening processes. By understanding their effect, one should be able to build models that better exploit improved knowledge representation. In Chapter 6, we present tools for detailed analysis and suggest how to use the results of the analysis to build better models.

The key result of the second part of the chapter, the comparison among measurement sets, is that a measurement set consisting of spectral measurements made both within and beyond segment boundaries can achieve significantly better performance than one consisting of only within-segment measurements Another important result of the comparison is that there appears to be little benefit to adding spectral measurements made at the beginning and end of a segment to a set of measurements made at the middle. As we discussed in Section 4.8.3, a plausible reason for this result is that the segment-based HMM does not need such measurements to model the spectral change within a phonetic region.

Finally, we compared our best phonetic recognition results with those reported in previous work. While several differences between our task and previous ones make a direct comparison difficult, the phonetic recognition accuracy we obtained is within the range of previously reported accuracies for systems of similar computational complexity. We do not believe that the tasks are different enough to alter our conclusion that the results reported here are competitive with those of other systems. Given the fact that there was little attempt to optimize the segmenter, the association rule between segment and phone label, the HMM topology, or the strategy for training biphones, we are very encouraged by these results. We are further encouraged by the result reported in Table 4.13, which suggests that the inclusion of biphone models does not hurt system performance, in spite of the small amount of training data available for these models.

If an attempt is made to optimize the above choices and to incorporate more sophisticated PDF and context modelling, we believe that phonetic recognition performance can improve substantially. At the same time, as stated in Chapter 3, we believe that the framework is a more convenient one in which to represent acoustic-phonetic knowledge. If this knowledge can be used to build models that are more robust than conventional models to factors such as speaker variability and a noisy environment, the segment-based HMM will be an attractive alternative for speech recognition.

Chapter 5

Word Spotting

While the phonetic recognition task presented in the Chapter 4 provides insight into the behavior of the system, it does not address issues that arise in building word models. Since speech recognizers generally recognize words and not phones, such issues are important. The main goal of this chapter is to explore three of these issues by assessing the system's performance on a word spotting task. The issues we explore are relevant for HMM word modelling, in general. However, the results we obtain will also provide insight into building word models within the segment-based HMM framework. Developing such insight is important if the segment-based HMM is to be applied to continuous speech recognition.

Another goal of this chapter is to introduce error diagnostic techniques and to show their utility. These techniques are extended in Chapter 6.

We investigate the following issues in this chapter:

1. the relative effectiveness of training a word model from data specific to the word vs. building one out of subword models,

2. the effect of the pronunciation network used to represent the word, particularly whether single- or multiple-pronunciation networks work better, and

3. the effect of measurement set on word spotting performance.

Before we discuss these issues and present experimental results, we will briefly describe the word spotting task and its relationship to the overall aims of the thesis. Then, the word spotter used in the thesis will be described in detail with particular attention being paid to the novel algorithms we use for scoring and performance evaluation. Finally, we present our results and discuss them. Within the discussion, we introduce an EDA technique we term the *segment score plot* for diagnosing errors and present a case study of its use.

5.1 The Word Spotting Task

In general, the task of a word spotter is to determine occurrences of one or more keywords embedded in other speech and/or noise. Applications include telecommunications [Wilpon 90, Wilpon 91, Chigier 92], gisting (determining the subject of discussion by spotting key words) [Rose 91, Rohlicek 92], and voice editing [Wilcox 92]. Finally, word spotting has also been used in a recognition system for identifying "islands of reliability" about which word string hypotheses are proposed [Kawabata 89].

In the present study, word spotting task is used as a test bed for investigating the three issues listed above while avoiding certain complexities encountered in building a continuous speech recognizer. Word spotting simplifies both computation and modelling. It requires less computation than recognition because it avoids the expensive search needed to determine a complete word string. Also, the task simplifies the modelling problem by allowing us to focus our modelling efforts on the words to be spotted rather than all words in the vocabulary. This contrasts with the word recognition paradigm, in which each word in the speech corpus being used to test the system would have to be modelled.

We simplify the task further by building a separate word spotter for each keyword to be spotted instead of a single spotter that identifies multiple words. Thus, the issues above can be investigated for each word individually to determine if the findings generalize over the complete set of words studied. Because we emphasize a data analytic approach to understanding system behavior, we look at a small number of words. These include function words that are short and often poorly articulated and are thus likely to be confused with fragments of continuous speech. Words of this sort studied include "what", "where", "near", "can" and "from.". We also looked at some content words that are frequent enough in the VOYAGER database to use for training and testing the spotter. These include "Harvard", "MIT", "nearest", and "Baybank.".

5.2 Algorithms for Acoustic Scoring and Performance Evaluation

5.2.1 Acoustic Scoring

The word spotter's task is to determine occurrences of one or more keywords $W_1, W_2, \ldots, W_V$ embedded in other speech and/or noise. Points where the word spotter indicates that some keyword has occurred are called *putative hits* [Rose 90] for that keyword. Putative hits can be classified as *correct detections* and *false alarms* depending on whether the keyword has occurred. Since in our work we build a separate word spotter for each keyword, we describe the scoring and evaluation algorithms for $V = 1$. In [Marcus 92], generalizations of these algorithms for particular applications and for the case of $V > 1$ are suggested.

Word spotting using HMM's involves defining an HMM for the keyword W and one or more HMM's for alternative input, be it speech or noise. As in [Wilpon 90], we refer to the latter as *garbage models*. The spotter determines the sequence of keyword and garbage models which forms the best acoustic match to the utterance. This determination is made by finding a path through a network of keyword and garbage models such as that illustrated in Figure 5.2.1 for the case of a single garbage and single keyword model, where a path is defined to be a sequence of states belonging to models in the network.

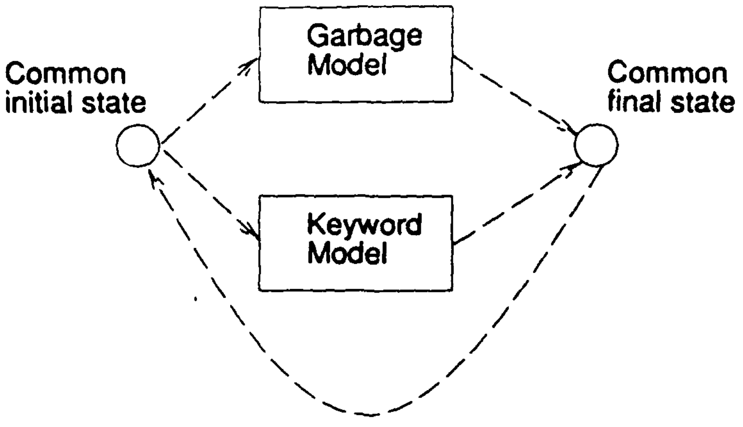


Figure 5.1: Keyword/garbage model network for a single garbage model and single keyword model. Dotted lines represent transitions between models. Details of topologies of keyword and garbage HMM topologies appear later in text. The illustrated network matches any sequence of keywords and garbage.

The illustrated network matches any sequence of keywords and garbage.

Most existing systems for word spotting use one of two algorithms to determine putative hits:

1. In [Rose 90, Wilpon 90, Wilpon 91], the Viterbi algorithm is used to find the best path through the network of keyword and garbage models. Segment sequences in this path which match the keyword model are labelled as putative hits.[1]

2. In [Rohlicek 89], Baum-Welch scoring [Rabiner 83] is used to determine the probability $\mathcal{W}(t)$ that the word ends at each segment t. Segments corresponding to local maxima of $\mathcal{W}(t)$ which exceed some threshold are considered putative hits. Putative hit starting segments are not determined.

[1] The existing systems described here are frame-based. We use the term *segment* to mean the unit upon which acoustic measurements are made.

The goal of our scoring algorithm is to compute the estimated probability $\mathcal{W}(t,\ell)$ that W begins at segment $t-\ell+1$ and is ℓ segments long, ending at segment t, given the sequence of observation vectors $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t$.[2] This is computed for $1 \le t \le T$, $1 \le \ell \le \ell_{\max}$, where $\ell_{\max}$ is a user-selected limit on the maximum number of segments which can correspond to word W and T is the total number of observed segments.

Let s_t be the state in the recognition network at segment t, and S_I and S_F be indices of the initial and final states of the keyword model, which is a left-to-right HMM. Using $P(.)$ to denote a probability density estimate, define the forward score $\alpha_t(i) = P(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t, s_t = i)$ as the estimated joint probability density of observing $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t$ and being in state i after segment t. Thus, $P(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t) = \sum_i \alpha_t(i)$. Then

$$\begin{aligned}\mathcal{W}(t,\ell) &= \frac{P(s_{t-\ell+1} = S_I, s_t = S_F, \boldsymbol{y}_1,\ldots,\boldsymbol{y}_t)}{P(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t)} \\ &\approx \frac{\alpha_{t-\ell+1}(S_I) P(\boldsymbol{y}_{t-\ell+2},\ldots,\boldsymbol{y}_t, s_t = S_F | s_{t-\ell+1} = S_I)}{\sum_i \alpha_t(i)}.\end{aligned} \tag{5.1}$$

The approximation is the usual one made when using Markov models, i.e., the distribution of the observations from time $t-\ell+2$ on is dependent only on the state observed at time $t-\ell+1$. The algorithm used to compute Eq. 5.1 is:

1. For all states i and for $1 \le t \le T$, compute $\alpha_t(i)$ using the Baum-Welch computation [Rabiner 83].

2. For $1 \le t \le T$, $1 \le \ell \le \ell_{\max}$, compute $P(\boldsymbol{y}_{t-\ell+2},\ldots,\boldsymbol{y}_t, s_t = S_F | s_{t-\ell+1} = S_I)$ by sequentially setting $s_{t-\ell+1} = S_I$ for all t and performing the Baum-Welch computation for $\ell_{\max}$ segments at each value of t. Thus, the keyword model is "slid" along the time axis.

[2] For notational simplicity, observation vectors will be expressed as column vectors rather then row vectors in this section.

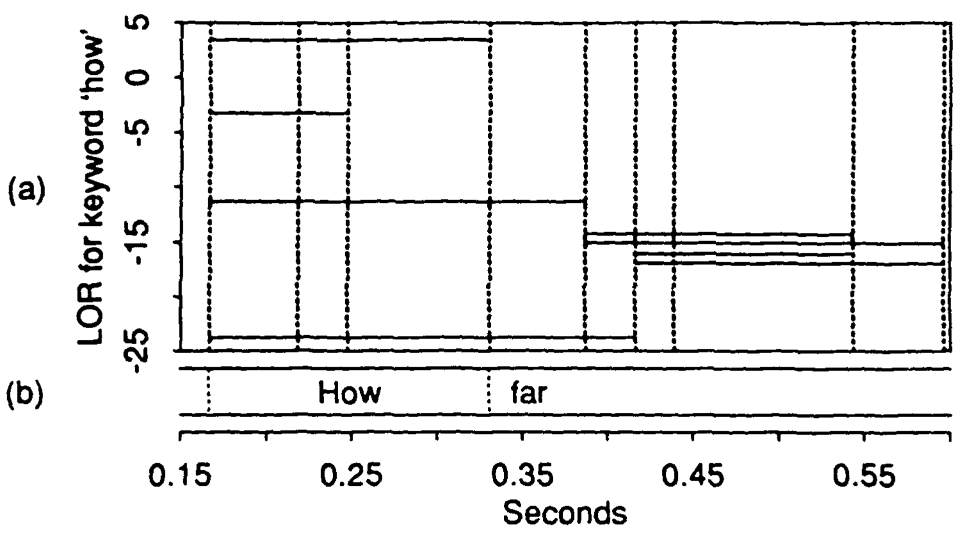


Figure 5.2: Log odds ratios for keyword "how." (a) LOR(t, ℓ) for each (t, ℓ) pair represented by a line segment at height LOR(t, ℓ) whose left and right endpoints co-occur with those of segments $t - \ell + 1$ and t. LOR's below -25 not included. Vertical dashed lines are segment boundaries. (b) Time-aligned orthographic transcription.

After performing these steps, Eq. 5.1 is readily computed. In the case of multiple keywords, Step 2 is repeated for every keyword W_v for $1 \leq v \leq V$ to produce scores $\mathcal{W}(t, \ell, v)$.

For displaying and storing the scores, we perform the monotonic transformation

$$\text{LOR}(t, \ell) = \ln(\mathcal{W}(t, \ell)/(1 - \mathcal{W}(t, \ell))$$

where LOR is an abbreviation of *log odds ratio.* This reduces the scores' dynamic range.

Figure 5.2 illustrates scores obtained in spotting the word *how*. The log odds ratio LOR(t, ℓ) is represented by a line segment at height LOR(t, ℓ) whose left and right endpoints co-occur with those of segments $t - \ell + 1$ and t. Thus, higher segments represent intervals most likely to be aligned to the keyword. Segment boundaries, which are represented by vertical dotted lines, are not equally spaced on the time axis because our system uses variable-length acoustic segments. Note that the highest scores occur near the keyword.

In our view, the major advantage our algorithm has over Viterbi decoding is that it yields keyword scores for *every* (t, ℓ) *pair* while Viterbi decoding only yields scores for segments where the keyword model is used in the best path. Thus, our algorithm provides more information for evaluating the system and we use this information in our performance measure, as we show in the next section. Additionally, the Baum-Welch score has been shown to yield better performance than the Viterbi score in certain applications [Schwartz 91].

The advantage of the algorithm over that of [Rohlicek 89] is that it yields scores for start-end segment pairs rather then for ending segments alone. This is important for error analysis since it allows both start and end points of false alarms to be determined.

Our algorithm requires more computation than that of [Rohlicek 89] due to the inclusion of Step 2 for recovering keyword starting points. However, at least for our system, the computation in Step 2 is small compared to that of Step 1.

In general, for a system that spots an arbitrary number of keywords, the computation in Step 1 is proportional to $T[(N_g + N_w + N_c) + M]$ where N_g, N_w and N_c are respectively the number of state transitions in the garbage models, the word models and between the two sets of models and M is the computation required to compute acoustic segment scores $P(\boldsymbol{y}_t)$.

The computation in Step 2 is proportional to $TN_w\overline{\ell_{\mathbf{max}}}$. where $\overline{\ell_{\mathbf{max}}}$ is the average over all words of the maximum number of segments to be considered for each word. Note that the acoustic segment scores computed for word model states in Step 1 can be reused in Step 2.

We should point out that in [Wilcox 92], another method is used for recovering both endpoints of the putative hit. Their method uses a heuristic to determine the most likely start segment for each putative end segment. Experiments to compare the accuracy and computational requirements of the two methods would be worthwhile but are beyond the scope of this work.

5.2.2 Performance Evaluation

In [Wilpon 91], it is suggested that there is some debate over how best to evaluate word spotters and that the controversy largely stems from the fact that each word spotter's requirements depends on the task for which it is used. Our approach to the problem is to use a performance measure dependent only on the spotter's ability to discriminate keywords from false alarms. The measure is essentially task-independent but can be easily tuned to particular tasks.

Word spotters have usually been evaluated by operating the spotter at putative hit thresholds that yield arbitrarily chosen false alarm frequencies, expressed as false alarms per hour per keyword (fa/h/w). The fraction of instances of the keyword correctly detected (the correct detection rate) is computed at each of these *operating points* and the average rate is used as an overall performance measure.

The measure proposed here is different in that test utterances are first divided into keyword and garbage *trials*. Keyword and garbage trial scores based on the log odds ratios LOR(t,ℓ) are computed. Garbage scores above a given threshold are designated as false alarms and keyword scores above that threshold as correct detections. The relationship between false alarm and correct detection rates as the operating point is varied is expressed as a *receiver operating characteristic* (ROC) and the area under the ROC is used as the performance metric.

Figure 5.3 depicts the process of determining trial scores from a corpus of transcribed speech. The algorithm is:

1. Allocate one keyword trial for each keyword token. For each trial k in the set of keyword trials determine begin and end segments B_k and E_k

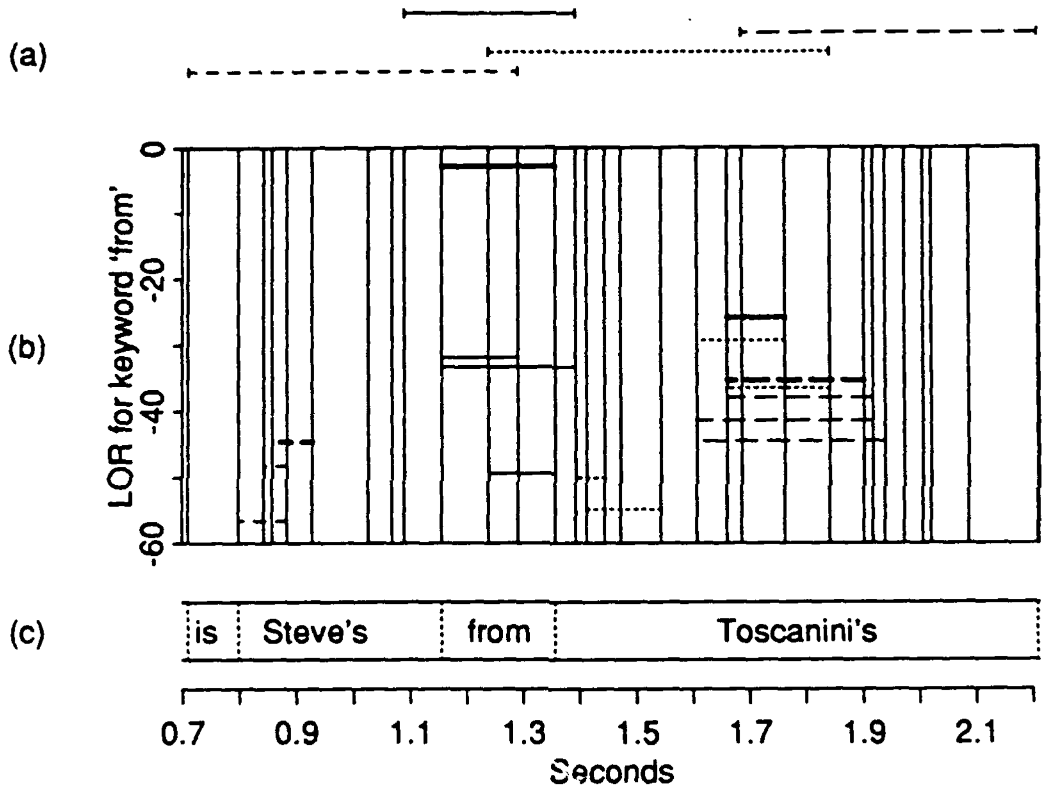


Figure 5.3: Trial score determination for keyword "from." (a) Trial intervals for $D_G = 12$, $\delta = 1$. Solid line represents keyword trial interval. Each dotted line represents a garbage trial interval. (b) Plot of LOR's. See Figure 5.2 for explanation. The line type used to represent LOR's associated with a particular trial corresponds to the line type used in (a) for that trial's interval. The thick lines represent LOR's corresponding to trial scores, e.g., the keyword trial score is approximately -2.0. For clarity, LOR's below -60 and some others have been removed. (c) Time-aligned orthographic transcription.

by setting $B_k = b_k - \delta$ and $E_k = e_k + \delta$ where b_k and e_k are the transcribed begin and end times associated with the token and δ is "padding" that accounts for imprecision in the transcription. We will refer to the (B_k, E_k) pair as a keyword trial interval (KTI).

2. Allocate to garbage trials intervals not associated with a keyword. In our system, each contiguous portion of the corpus between KTI's is divided into garbage trial intervals (GTI's) of D_G segments where D_G is chosen to be roughly the maximum duration of the keyword observed in the training set. GTI's adjacent to KTI's or to the starts and ends of utterances may be truncated to fewer than D_G segments. We motivate our choice of D_G below. We allow GTI's to extend into KTI's to catch false alarms that occur when the end of a keyword forms the beginning of a false alarm and vice-versa. Our current system also allows overlaps between adjacent garbage intervals but this is of little importance and will not be discussed further.

3. For each keyword trial k associate all LOR(t, ℓ) such that $B_k \leq t - \ell + 1 \leq t \leq E_k$, i.e, the log odds ratios for all begin-end segment pairs completely contained within the keyword's trial interval. The keyword trial score X_k is the maximum value of LOR(t, ℓ) associated with the trial. The maximization removes the overlapping putative hits associated with a single trial.

4. Associate each remaining LOR(t, ℓ) with the garbage trial g whose interval's center is closest to $t - \ell/2$, the center of the interval spanned by LOR(t, ℓ). Remove from consideration any LOR(t, ℓ) which overlaps an interval exclusively associated with a keyword trial since it may be large due to a keyword occurrence rather then due to a false alarm. The garbage trial score Y_g for each garbage trial g is the maximum LOR(t, ℓ) associated with the trial.

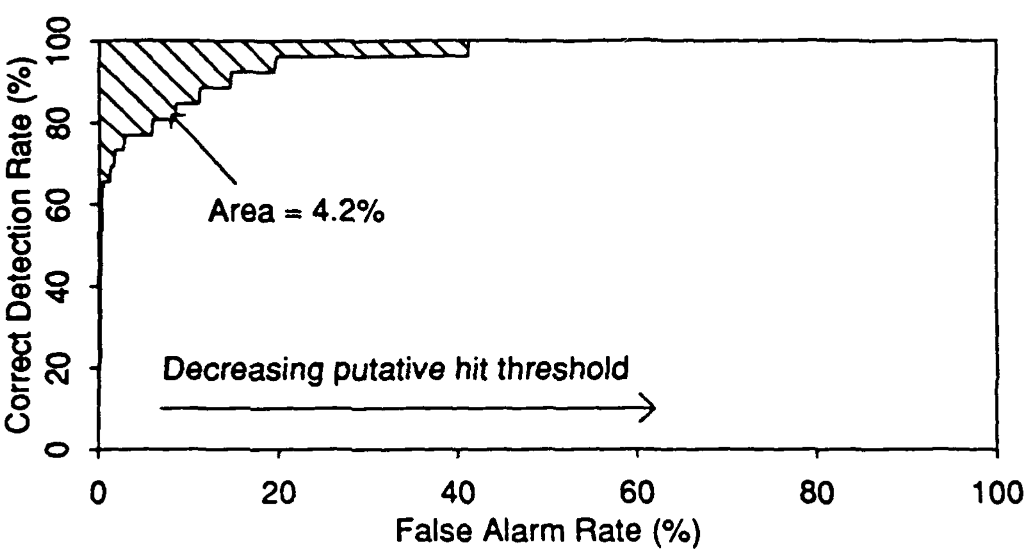


Figure 5.4: Receiver operating characteristic. Rates are expressed as percentages. The area over the curve is indicated by shading. There were 26 keyword trials and 711 garbage trials.

We have set D_G to be near the maximum keyword duration because this is long enough so that trial scores can be considered as independent, i.e., it is unlikely that a portion of an utterance that acoustically matches the keyword will cause high scores on more than one garbage trial due to overlapping $\mathrm{LOR}(t, \ell)$ that get assigned to different trials. At the same time, D_G is short enough that distinct false alarms will all be considered in the evaluation instead of being collapsed into a single trial score.

Once trial scores have been computed, an ROC is constructed. Figure 5.4 illustrates a sample ROC. In the figure, the area A_o over the ROC is shaded. The area under the ROC, which is used as a performance metric, is $1 - A_o$. The ROC represents the relationship between the false alarm rate $F(\eta)$ and the correct detection rate $C(\eta)$ for a set of values of the putative hit threshold η. To produce as smooth an ROC as possible, η is set to each of the garbage trial scores.

Formally, let $Y_{(i)}$ be the i^{th} smallest value of the set of garbage trial scores $\boldsymbol{Y}$, N_G and N_K be the numbers of garbage and keyword trials, and $\boldsymbol{X}$ be the set of keyword trial scores. For each i a point (F_i, C_i) on the ROC is produced such that

$$F_i = 1 - (i-1)/N_G \text{ and } C_i = \|\{X_k \in \boldsymbol{X} | X_k > Y_{(i)}\}\|/N_K.$$

where $\| \ \|$ is the cardinality operator. Thus, F_i is the fraction of garbage trials whose scores are at least as great as $Y_{(i)}$ and C_i is the fraction of keywords whose trial scores are at least as great as this value. The points $(F_0 = 1, C_0 = 1)$ and $(F_{N_G+1} = 0, C_{N_G+1} = 0)$ are added to the ROC as well, representing $\eta = \mp\infty$.

One attractive feature of this type of ROC is that false alarm and false rejection rates are similar to recognition error rates in the sense that they are expressed as fractions rather then errors per unit time. Thus, they may be easier to interpret by the speech recognition community. In particular, the false alarm rate can be interpreted as being the error rate over all garbage trials and the false rejection rate can be thought of as the error rate over all keywords. If each GTI is about the same duration as a typical keyword, the sum of the two rates at a particular η, perhaps weighted by the relative frequency of garbage and keyword trials, would be similar to a word recognition error rate.

Once the ROC is determined, the correct detection rate can be averaged over one or more operating points to determine a performance measure analogous to the (fa/h/w) measure. Such a measure is dependent on both the spotter's ability to discriminate keywords from false alarms and on the choice of operating points.

In our work, we prefer to use a measure that is independent of operating point since we are interested solely in the spotter's discrimination ability. Thus, we use the area under the ROC as a performance metric, a larger area corresponding to better performance. The area, denoted A, is computed using the trapezoid rule:

$$A \;=\; \sum_{i=0}^{N_G} \frac{1}{2}(F_i - F_{i+1})(C_i + C_{i+1}) = \frac{1}{N_G}\sum_{i=1}^{N_G} C_i$$

Note that this can be interpreted as the correct detection rate averaged over *all* distinct operating points. It therefore makes use of all of the information provided by the ROC, unlike any measure based on arbitrarily chosen operating points. Additionally, the area under an ROC can be shown to be an estimate of the probability that given a randomly chosen keyword trial and a randomly chosen garbage trial, the spotter will correctly classify the two trials [Green 66]. Thus, the area can be interpreted as a type of error rate.

Another feature of this performance measure is that there exists a non-parametric significance test for comparing the area under several ROC's produced for the same set of trials [Delong 88]. The test can be used to compare word spotters on the same task much as McNemar's test is used to compare recognizers [Gillick 89]. We present without derivation the test for comparing two spotters, due to [Delong 88], which includes a detailed derivation and extends the test to the case of an arbitrary number of ROC's. Significance levels referred to in our work are computed using this test.

Because identical test data are passed to the two spotters, trial intervals for evaluating the spotters are identical. Let X_{ki} be the score on keyword trial k for spotter i with $i = 1, 2$ and $\boldsymbol{X}_i$ be the set of these scores. Similarly, let Y_{gi} be the score on garbage trial g for spotter i and $\boldsymbol{Y}_i$ be the set of these scores. Let

$$Q_{ki} = \frac{1}{N_G}\|\{Y_{gi} \in \boldsymbol{Y} | Y_{gi} < X_{ki}\}\| \text{ for } 1 \le k \le N_K$$

and

$$R_{gi} = \frac{1}{N_K}\|\{X_{ki} \in \boldsymbol{X} | X_{ki} \ge Y_{gi}\}\| \text{ for } 1 \le g \le N_G$$

for $i = 1, 2$. Finally, let

$$z_{ij} \;=\; \frac{1}{N_K(N_K - 1)}\sum_{k=1}^{N_K}(Q_{ki} - A_i)(Q_{kj} - A_j) +$$

$$\frac{1}{N_G(N_G-1)}\sum_{g=1}^{N_G}(R_{gi}-A_i)(R_{gj}-A_j)$$

for $i = 1,2$ and $j = 1,2$. Then, assuming independence of trial scores for a given spotter, under the null hypothesis that the areas under the ROC A_1 and A_2 are equal for both spotters, $A_1 - A_2$ is asymptotically normal with mean 0 and variance $z_{11} - 2z_{12} + z_{22}$. Thus, a t-test with the statistic

$$(A_2 - A_1)/\sqrt{z_{11} - 2z_{12} + z_{22}}$$

can be used as a significance test. Note that the test only depends on the trial scores' order statistics, not on their distributions.

For a test set with a small number of speakers, as ours is, this test should not be applied because the assumption of independence among trials is a poor one. This issue was discussed in Chapter 4 in the context of phonetic recognition.

5.3 Word- vs. Subword-trained Models

One choice faced in the design of a speech recognition system is the selection of an inventory of lexical-acoustic units. As discussed in Section 1.3, in an HMM recognizer each lexical-acoustic unit consists of an HMM which stores a parametric representation of the estimated PDF of the sequence of observation vectors associated with the unit's label. The parameters are determined by collecting statistics of the segment sequences associated with that label in the recognizer's training set.

In Chapter 4, we used phone and biphone lexical-acoustic units since the recognition task was to recover the utterance's phone string. However, in a word recognition or spotting task, models for words must be built. The most straightforward way of building such models in an HMM system is to train the word model from tokens of that word. We refer to such models as *word-trained models.* Alternatively, each word can be represented as a sequence or network of subword units, with the sequence or network specified by a lexicon, as discussed in Section 1.3. Then the HMM for each word can be built by connecting the final state of the HMM for each subword to the initial state of the HMM of the subword that follows it in the lexicon's representation for the word. We refer to word models built this way as ***subword-trained models.*** If each subword is a phone, we refer to the models as ***phone-trained models.***

The drawback of word-trained models is that in a large vocabulary recognizer it is difficult to obtain enough training data for each word to make accurate parameter estimates of the observation sequence PDF's. Thus, word models tend to be used in small-vocabulary systems where it is feasible to collect many instances of each word, e.g. [Rabiner 83].

In large vocabulary recognizers, phone-trained models have been used, e.g., [Jelinek 76]. The problem with this approach is that the acoustic realization of a phone can depend on the identity of nearby phones and the stress placed on the syllable in which the phone occurs. We refer to these factors as the phonetic and prosodic context, respectively. Due to these effects, the PDF for the observation sequence associated with a phone conditioned on its context may differ considerably from the unconditional PDF estimated from all data associated with the phone.

Various systems have employed word models built out of inventories of lexical-acoustic models intermediate between phone and word models, such as diphone models [Colla 85, Colla 86, Vicenzi 86]. Other systems have used distinct phone models to represent the phone in different environments. This approach has been used to model the effect of left and right phone identity [Schwartz 84, Schwartz 85, Paul 88, Lee 90] and of word context [Chow 86, Lee 88] on acoustics. The resulting models are referred to as triphone and word-specific phone models. Finally, distinct vowel models have been used to account for the effect of syllable type on vowel duration [Deng 89].

These unit inventories are more sensitive to context than are phone models but are harder to train reliably because there are less training data for each phone in a particular context than for the phone in all contexts. In systems which use triphone or word-dependent models, the lack of training data has been compensated for using two methods: interpolation and clustering. Parameters of interpolated models are weighted averages of context-dependent and context-independent model parameters. The weighting can be done either manually [Schwartz 84, Schwartz 85, Chow 86] or automatically [Lee 90]. Clustered models are built by averaging model parameters for different contexts together into single models, thus sharing data among the contexts. This, too, can be done manually, by pre-determining which contexts should be combined [Deng 88, Deroualt 88], or automatically, using various similarity measures for clustering [Paul 88, Lee 88]. At this time, many state-of-the-art recognition systems, e.g., [Chow 87, Cohen 90, Paul 91] use clustered triphone models as their units and these have been shown to outperform either phone models or unclustered triphone models [Lee 88].

For a more comprehensive review of previous approaches to the unit selection problem, see [Lee 90].

It is beyond the scope of our work to compare a wide variety of the above modelling strategies. Thus, we confine ourselves to comparing spotter performance of word-trained keyword models to subword-trained models consisting phone and biphone HMM's trained as discussed in Chapters 3 and 4. The motivation for these experiments is to develop some insight into the performance of the two types of models within our system rather then to make new contributions to the problem of unit selection.

We postpone detailed description of the experimental procedure and results of this comparison to Section 5.5. The experiments examine the effect of the pronunciation network in conjunction with the mode of training the word models. Thus, we outline the pronunciation network issue in the next section before presenting experimental results.

5.4 Pronunciation Networks: Single vs. Multiple Pronunciations

5.4.1 Previous Work

To build a subword-trained model for a given word, the allowable sequences of subwords that compose the word must be specified with a pronunciation network. Such networks are generally built by beginning with a dictionary of a single or a few *baseforms* for each word to be modelled. The baseforms specify either phoneme strings, e.g., [Cohen 74], or phone strings, e.g., [Lee 88] that are determined from an existing dictionary, e.g., [Cohen 74] or from the system designer's intuition, e.g., [Weintraub 87]. Then, in some cases, phonological rules are applied to convert the baseforms into the final pronunciation networks. Such rules would convert the phoneme /t/ into the phone /ɾ/ (flap) in some contexts, for example. Rule systems of this sort are described in [Cohen 74, Weintraub 87, Rudnicky 87].

As discussed in Section 1.1, both single- and multiple-pronunciation networks have been used in continuous speech recognizers. Lee [Lee 88] found that single-pronunciation networks performed as well as multiple-pronunciation networks. SRI researchers [Weintraub 89, Cohen 90] found the opposite to be true. Weintraub et al. [Weintraub 89] attributed the difference between the two results to the fact that the networks used in their work allow relatively few pronunciations per word compared to those of the multiple-pronunciation networks used by Lee, and thus provide a more appropriate degree of constraint on allowable pronunciations [Weintraub 89]. These are the only two published comparisons of single- and multiple-pronunciation networks of which we are aware.

5.4.2 Current Methodology

In the experiments to be described below, we compare word spotting performance for keyword models that allow single and multiple pronunciations. As we show below, the networks for each word are based on relative frequencies of the phonetic transcriptions observed for the word in a set of training data. As described in Chapter 2, the phonetic transcription for each utterance in the corpus was determined automatically by first expressing each utterance as a network of phone labels determined by the application of phonological rules [Zue 90a] to each word in the utterance. Then, the SUMMIT recognizer [Zue 89a] was used to determine the sequence of phones tha+ provided the best acoustic match to the utterance over all those allowed by the network. The automatic transcription was checked and sometimes modified by a transcriber. Thus, the relative frequencies of the phonetic transcriptions for each word and hence the pronunciation networks used in our work are influenced by the phonological rules, the behavior of the SUMMIT recognizer and the judgment of the transcriber. We built the pronunciation networks this way because the methodology was convenient to implement and adequate for the purposes of comparing single- and multiple-pronunciation networks, not because we advocate such a complicated approach to the problem in general.

5.5 Experiments on Unit Selection and Pronunciation Networks

5.5.1 The Basic Experimental Procedure

In this section, we describe the experiments and results obtained in experiments for which we compare word- and subword-trained keyword models represented by several different pronunciation networks. For each keyword examined the procedure was the same: a set of garbage models was built and used throughout the experiments. The model for the keyword was varied and spotter performance evaluated as a function of the keyword model used. The area over the receiver operating curve was used as a performance metric, as described in Section 5.2.2. A separate spotter was built for each keyword investigated. The test set used consisted of the 232 VOYAGER utterances from the five male speakers listed in Table 4.6.

In applying the performance evaluation metric for each word, we had to determine the length of the garbage trial intervals, D_G, and had to specify keyword trial intervals. For each keyword, D_G was set to the maximum number of segments associated with the keyword over all tokens of the keyword in the training set. In making this computation, a segment was deemed to be associated with the keyword if at least half of it overlapped the aligned orthographic transcription region for the token. The parameter is tabulated for each word in Table 5.1, which appears in Section 5.5.3.

A keyword trial was allocated for each token of the keyword observed in the test set and also for each token of a word of which the keyword is a root. For instance, each token of the word "where's" was associated with a keyword trial interval for the keyword "where." Figure 5.3 of Section 5.2.2 provides an example of keyword and garbage trial determination for the keyword "from." The first segment of the keyword trial interval for a keyword token that begins at time b is the one preceding the segment whose beginning and end times bracket b. Thus, if β_j and ϵ_j, the beginning and end times of segment j, respectively, satisfy the inequality $\beta_j \leq b \leq \epsilon_j$, segment $j-1$ is set to be the first segment of the interval. The segment preceding j is used rather then j itself to allow for the imprecision in the transcription process. The final segment of the interval is determined in a similar fashion.

For all experiments reported in this section, the baseline measurement set described in Section 4.2 was used. The set consists of seven hair-cell envelope principal components measured at the middle frame, seven hair-cell envelope principal component differences measured across the segment's right edge, and duration.

5.5.2 The Garbage Network

Two types of garbage networks are used in existing word spotting systems. We will refer to these as *unlabelled* and *labelled* networks. An unlabelled network is a single HMM trained using all non-keyword input. Labelled networks are trained by labelling non-keyword speech, building different HMM's for speech associated with different labels and connecting the HMM's.

Evidence from previous work suggests that the relative performance of the two types of network depends on the situation. For instance, in [Wilpon 90], word spotting was used in a telecommunications application to detect keywords embedded in sentences. Certain words showed up frequently in the non-keyword speech. There was little performance improvement when models of these words were incorporated in an otherwise unlabelled garbage network that consisted of an HMM that used a mixture Gaussian PDF. In [Rohlicek 89] an unlabelled garbage network of this type was used as well for a task of spotting one of twenty keywords in continuous conversational speech.

Conversely, Rose and Paul [Rose 90] compared performance obtained with unlabelled, phone-labelled and triphone-labelled garbage networks and obtained the best results from the phone-labelled network. In that case, the task was to detect keywords in spontaneous, noisy speech of greater variety than that of [Wilpon 90].

For our experiments, we chose to use a labelled garbage network in which the models are the set of phone and biphone models introduced in Chapter 3 and used in the phonetic recognition experiments of Chapter 4. We made this choice because it is easier to analyze spotting errors if the garbage model is labelled since the phonetic confusion responsible for each error can be determined.

Thus, the garbage network used is almost exactly the network of phone and biphone HMM's used in the phonetic recognition experiments of Chapter 4. The resulting keyword/garbage network is illustrated in Figure 5.5. In this example, the spotted word is "near."

The network was modified slightly for use in the word spotter. First of all, because the garbage model is supposed to characterize the statistics of non-keyword speech, all instances of the keyword were removed from the training

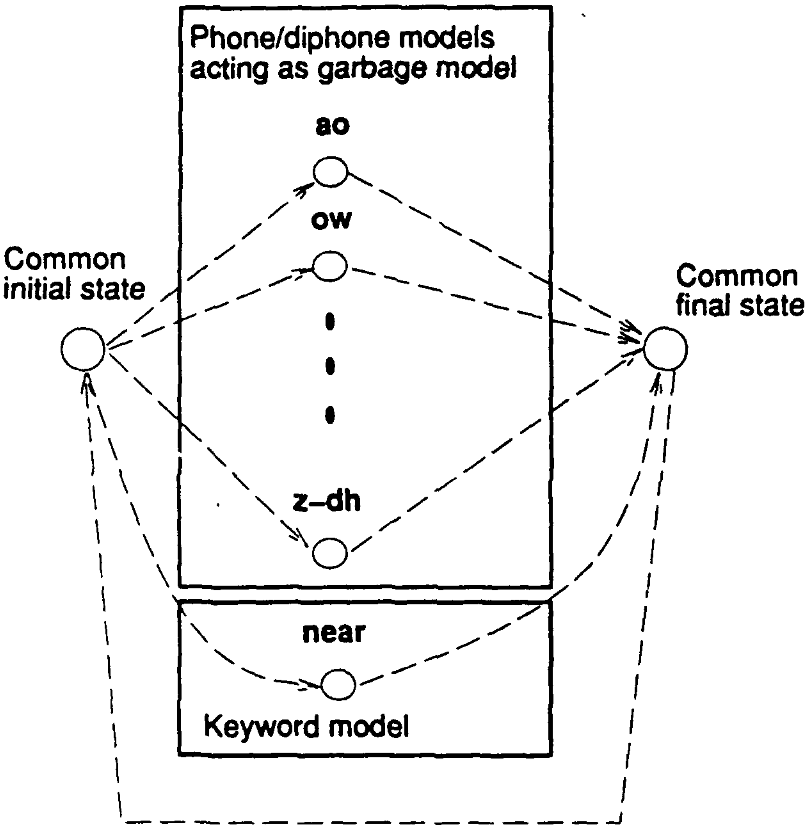


Figure 5.5: Keyword/garbage network for word spotting experiments. In this example, the spotted word is "near." Final model states are connected to initial model states. This is represented in the figure with common initial and final states. However, since a bigram model is used, the arc between the final and initial common states actually represents distinct arcs between every pair of models, each with a distinct transition probability.

set before training the phone and biphone models. Thus, for example, for the keyword "where", all segment sequences associated with /w/ occurring in "where" were removed from the training set before training the **w** garbage model. Specifically, a phone or biphone segment sequence was removed if 1/4 or more of any segment in the sequence overlapped with the transcribed keyword interval. The value of 1/4 was determined empirically by noting that when lower values were used, segment sequences were removed that overlapped with the keyword due to a slight misalignment between transcription and segmentation boundaries rather then because they were truly associated with the word. When higher values were used, sequences that truly were associated with the word were left in the training set. We should point out that while it is theoretically correct to remove segment sequences associated with each keyword from the garbage model training set, the number of such sequences was small compared to the complete training set and so their removal might not have had much practical effect on word spotter performance.

The second difference between the garbage network and the phonetic recognition network of Chapter 4 is that transition probabilities between models in the combined keyword/garbage network must take the keyword into account. For example, for spotting "where", instances of /w/ being followed by /ɛ/ in the "where" were not used in computing transition probabilities between the corresponding phone models. The removal of these sequences was accomplished simply by replacing all phonetic labels in the transcription that are associated with the keyword with the keyword itself before transition probabilities were computed. For example, the transcription /**tɛlmiwɛrɪɾɪz**/ of the phrase "tell me where it is" was modified to become /**tɛlmi**[where]**ɪɾɪz**/ where the brackets are used to denote that the enclosed label is a word rather then a phone. After this transformation, bigram transition probabilities among phone and biphone models were computed as described in Appendix B. Finally, transition probabilities into and out of the keyword must be computed. In principle, bigram probabilities could be computed with the modified transcription using the algorithm described in Appendix B, i.e., the keyword could be treated just as the phones and biphones are treated in the computation. In that case, the transition probabilities into and out of the keyword would be dependent on the previous and following labels, respectively. However, so as to concentrate on issues of acoustic modelling, we chose not to model the dependence of key word occurrence on surrounding phonetic context since this is really a language modelling issue. Thus, the transition probability into the word was set to be the estimated probability that the word follows any label. Similarly, the transition probability from the keyword model to each phone or biphone model was set to the estimated probability that the phone or biphone label follows any other label. Finally, in accord with the algorithm described in Appendix B, the probability of the keyword beginning the utterance was set to be the fraction of utterances in the language training set that begin with the keyword.

5.5.3 Single-Pronunciation Networks

The first set of experiments we ran compared word-trained keyword models to subword-trained keyword models that allowed a single pronunciation. The pronunciation used was that which occurred most frequently in the pronunciation training set. This set included 473 utterances from the ten male VOYAGER speakers used to train the subword models as well as 780 utterances from sixteen female VOYAGER speakers.

Training Subword-Trained Models

Each of the single-pronunciation subword-trained models included only phone models, not biphone models. Figure 5.6 displays an example of such a model, which was constructed by stringing together the **n**, **ih** and **axr** phone models.

The topology and training method for each phone model was that discussed in Chapter 3 and each model used the diagonal Gaussian PDF representation discussed in Chapter 4. Each phone model used to build a word model was

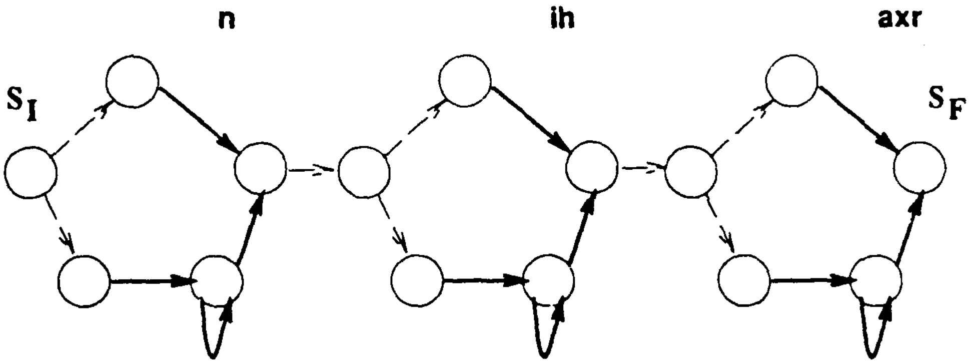


Figure 5.6: Single-pronunciation topology for "near." Solid lines are transitions which match an acoustic segment. Thin dashed lines are null transitions, for which no segment is matched. Subword model names included in figure. Initial and final word model states marked S_I and S_F, respectively. The same topology is used for both word- and subword-trained models.

trained by all instances of that phone in the subword model training set, which consisted of 2150 TIMIT utterances and 473 VOYAGER utterances from male speakers, as discussed in Chapter 3. Thus, for example, the **n** model used in the model for "near" was trained from all instances of /n/ in the training set. As shown in the figure, each word model was constructed by connecting the final state of each phone model in the pronunciation to the initial state of the following phone model. Table 5.1 enumerates the keywords used, the most common pronunciation for the word in the pronunciation training set, the number of instances of each keyword in the model training set, and the value of D_G (in segments) used in the performance evaluation of each keyword spotter.

Training Word-Trained Models

The word-trained model for each keyword was trained from all tokens of the keyword observed in the training set. For each such token, the segment sequence used to train the model was set to be that whose beginning and end

Keyword	Most common pronunciation	N_T	D_G
Harvard	**hh a r v axr dcl**	88	13
MIT	**eh m ay tcl t iy**	67	15
Baybank	**bcl b ey bcl b ae ng kcl k**	26	16
from	**f axr m**	207	12
where	**w eh axr**	88	10
what	**w ah tcl**	56	9
can	**k ix nx**	77	14
near	**n ih axr**	48	9
nearest	**n ih axr ix s**	44	12

Table 5.1: Keywords for word spotting experiments. Most common pronunciation was determined over the pronunciation training set. N_T is number of tokens of the keyword in model training set. D_G expressed in number of segments.

times most closely matched the beginning and end times of the token's interval according to the aligned orthographic transcription. Each word-trained model employed the same topology as its subword-trained counterpart. This was done so that different topologies between the two models would not be confounded with the training mode in interpreting results.

As discussed in Section 3.7, the particular phone model topology employed made training each phone model straightforward in that the position of each state in the model uniquely determines the segments used to train it. The uniqueness is due to the fact that there is only one self-looping state in each phone model. However, as can be seen in Figure 5.6, the same is not true for the word model. Thus, we used the segmental K-MEANS algorithm to train each word model. The algorithm is described in detail in [Rabiner 89]. Briefly, the algorithm is:

1. Use the subword-trained model as a "seed."

2. For each training token, compute an alignment between the segment sequence and the states in the model using the Viterbi algorithm. The alignment so determined maximizes the segment sequence's acoustic

score on the model over all allowable alignments. The alignment associates each segment with a state in the model sequence. Thus, after each token is so aligned, each state has a set of segments associated with it.

3. Update the mean and weight vectors for each state in the model based on the statistics of the training segments associated with that state. The update formulas are specified in Equations 4.9-4.11. If a state has fewer than two segments associated with it, the sample variance of each measurement over the segments associated with the state is undefined and thus so are the weights. These states are not updated and so retain the values specified by the seed model.

4. Update the transition probabilities based on the set of state sequences determined by the alignment algorithm. Specifically, the transition probability a_{ij} between states i and j in the model is given by $a_{ij} = N_{ij}/N_i$ where N_{ij} is the number of times that state j follows state i in the set of state sequences and N_i is the number of times state i is visited over all the state sequences.

5. Go to Step 2.

The algorithm was iterated five times for each keyword. The acoustic match score achieved by the Viterbi alignment tended to increase for the first few iterations before levelling off, indicating that the algorithm had converged to a locally optimal set of alignments.

Results

The results comparing word- and subword-trained models are tabulated in Table 5.2. There are two points worth noting. First of all, performance is very poor for some words, especially given the error rate interpretation of the area over the ROC given in Section 5.2.2. Also, the spotter appears to perform much

	Area Over ROC (%)		
	Subword-trained	Word-trained	N
Harvard	14.3	6.3	20
MIT	17.8	34.8	26
Baybank	30.9	18.2	14
from	8.1	4.2	60
where	8.3	8.2	60
what	47.4	29.9	56
can	44.4	20.8	20
nearest	7.5	0.2	17
near	8.3	0.8	29

Table 5.2: Word- vs. subword-trained models: Single-pronunciation networks. N denotes the number of keyword tokens in the test set.

worse for content words such as "MIT" and "Baybank" than it does for some of the function words, such as "from" and "where." This is surprising because it is likely that there are more instances of speech fragments confusable with the shorter and less well-enunciated function words than there are confusable with the content words. We look more closely at this phenomenon below. The other interesting result is that word-trained models outperform subword-trained models for all words with the exception of "MIT."

Of course, this is a function of the particular training set used in the experiments. If there were very little word-specific data for each word, then presumably the subword-trained models would prevail due to the greater amount of training data available to them. This can be seen in Figure 5.7, which plots performance for both types of models over a range of training set sizes for five of the keywords in the study.[3] We selected only five words for this study to limit the required computation. For word-trained models, the data point for each training set size was produced by drawing a random sample of

[3] There are small differences between the results displayed on these plots and those cited in Table 5.2 due to minor differences in the performance evaluation algorithms used to produce the two sets of results. In particular, the plots were produced from an older version of the algorithm. The algorithm used to produce the results cited in the table was used to generate the results reported throughout the remainder of the chapter.

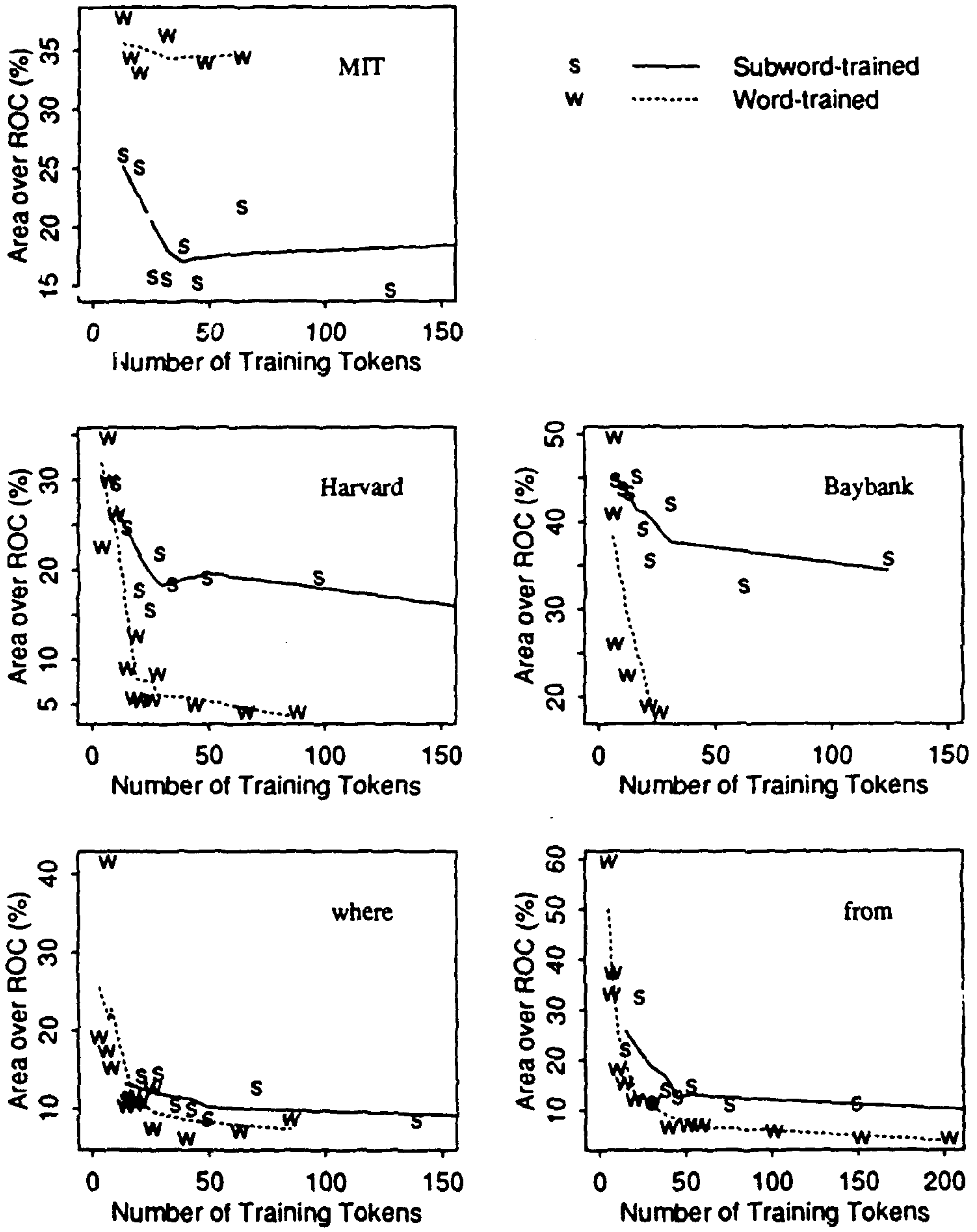


Figure 5.7: Effect of training set size on performance of word- and subword-trained models. Process for generating data points described in text. Lines through points produced by S-Plus routine supsmu.

keyword training tokens of that size from the complete training set and using the sample to train the keyword model. For subword-trained models, each data point was produced by specifying a fraction of the complete training set and drawing a random sample of that size for use in training the subword models. Since with this scheme, training set sizes varied among the subword models, the co-ordinate representing the number of training tokens was set to be the minimum number over all subword models. This choice was made on the assumption that model performance would be most effected by the model's "weakest link", i.e., the subword with the least training.

An alternative to this approach would have been to provide each subword model with the same number of training tokens. However, the scheme we implemented provides a more realistic simulation of different subword model training set sizes since it preserves the relative training set sizes for each subword available in the complete training set.

We only display results for subword-trained models trained from 20% or less of the complete training set. Subword-trained performance did not improve appreciably when more data was used. The lines drawn through the data points were produced by the S-Plus scatterplot smoother routine **supsmu**. The routine fits an estimate of the mean y value of the data as a function of x. Thus, in this case, the line on each graph represents a smoothed estimate of the area over the ROC as a function of the number of training tokens.

From these plots, it can be seen that if there is a very small number of training tokens for a given word, then a subword-trained model trained from the complete training set will usually perform better. For instance, when fewer than about ten tokens are used to train the "from" model, the area over the ROC exceeds 20%, as compared to about 12% for the subword-trained model that uses all the available training data. However, the number of training tokens required for word-trained models to prevail is apparently quite small. In fact, for all words in this figure but "MIT", word-trained models trained with about 25 or more tokens outperform subword-trained models trained from the complete training set. This result is further evidence of the importance of phonetic context modelling.

5.5.4 Multiple-Pronunciation Networks

Single-pronunciation networks of subword models fail to account for two sources of variability in the sequence of subword models that can match a given keyword. First of all, due to phonological variability, distinct instances of a given word might be composed of distinct phone sequences. Second of all, due to segmenter variability a given phone sequence might be best represented acoustically by more than one model sequence. For example, the sequence /ði/ might be represented by the two-model sequence **dh iy** or by the single biphone model **dh-iy**. Related to this issue is the fact that because each subword model within a single-pronunciation network must match at least one acoustic segment, the minimum number of segments that can match the single-pronunciation network is the length of the phone sequence used to represent the word. For example, from Figure 5.6 it can be seen that the single-pronunciation model for "near" admits only sequences of at least three segments. This is true of both subword- and word-trained models since the two have the same topology. In this section, we show that removing these restrictions leads to a large performance increase for many of the keyword spotters. In particular, we consider two types of multiple-pronunciation keyword models, which we term *full networks* and *skip networks*.

Full Networks

Full networks account for both phonological and segmenter variability. Each such network was built in a two-stage process. In the first stage, the pronunciations occurring in the pronunciation training set were used to derive a network of phone model labels. We term this the keyword's *phone network*. Figure 5.8 illustrates the phone network for "near." Transition probabilities are printed above each arc.

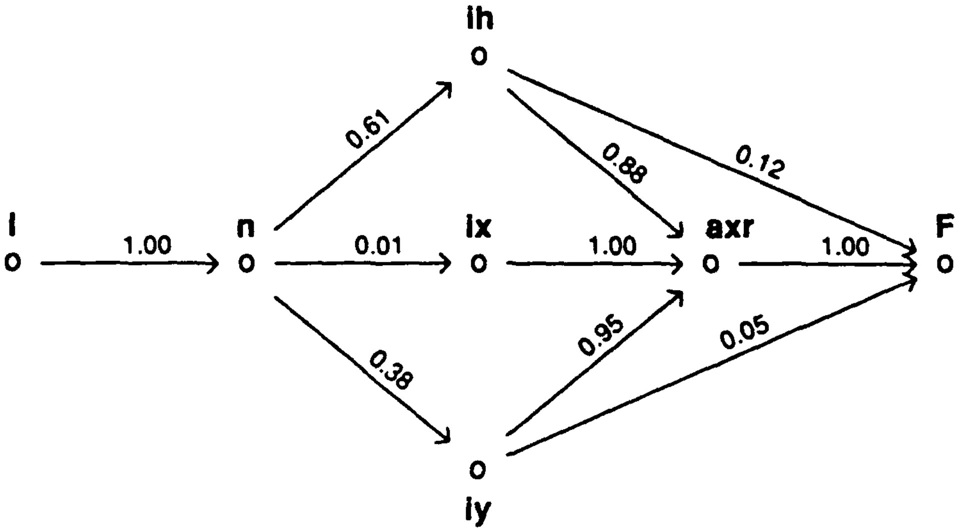


Figure 5.8: Phone network for "near." Word-initial and word-final nodes labelled I and F. Numbers represent transition probabilities.

The algorithm used in the first stage is very simple except for cases in which there are instances of a multiply occurring phone in the pronunciation of the word. In our presentation of the algorithm, we enclose in brackets the special steps required for this case.

1. Allocate a network node for each phone label occurring at least once in the pronunciation of the word. Label each node with the corresponding model label. Also allocate special word-initial and word-final nodes.

1a. [If a phone label occurs more than once in some pronunciations (e.g., the **b** in "Baybank"), allocate as many distinct nodes with this label as the maximum number of occurrences of the label observed in any pronunciation, distinguishing them with distinct indices. (e.g., allocate nodes **b1** and **b2** in the "Baybank" network.)]

1b. [For each pronunciation in the training set with a phone whose label occurs more than once, re-express the pronunciation using a distinct label for each occurrence of the multiply occurring phone. For instance, if one token's pronunciation of "Baybank" is **bcl b ey bcl b ae ng kcl k**, rewrite it as **bcl1 b1 ey bcl2 b2 ae ng kcl k**. Each label index refers to the role that the label plays in the word's pronunciation, not necessarily to its order of occurrence in a particular token. For instance, in this example **b2** refers to the /b/ that occurs before the /æ/ in "Baybank," not necessarily to the second /b/ in the token. Thus, if the pronunciation were **bcl ey bcl b ae ng kcl k**, indicating that the first /b/ burst was not labelled, it would be rewritten **bcl1 ey bcl2 b2 ae ng kcl k** and the first occurrence of /b/ would be labelled **b2**. The pronunciations were rewritten manually.]

2. Compute transition probabilities q_{AB} between every pair of nodes A and B. Set $q_{AB} = N_{AB}/N_A$ where N_{AB} is the number of pronunciations in which node B follows node A and N_A is the number of pronunciations in which node A occurs.

The second stage of the algorithm for building full networks augments the phone network with nodes for biphone models. Figure 5.9 illustrates the full network for "near" that results from the algorithm.

In the second stage, for each pair of sequentially occurring phones for which there is a biphone model a node representing the biphone model is added to the network. For example, if node A is followed by node B with a non-zero probability and there exists a biphone model labelled A–B, then a node must be added to the network. For instance, the sequence **eh axr** in the model for "near" illustrated in Figure 5.8 necessitates adding a node for the biphone model **eh-axr** to the network, as can be seen in Figure 5.9. The transition probability q_{CA} for any node C that can be followed by node A must now be split among nodes A and A–B. The splitting formula can be expressed as

$$q_{CA} = \hat{q}_{CA} + \hat{q}_{C,A\text{-}B} \tag{5.2}$$

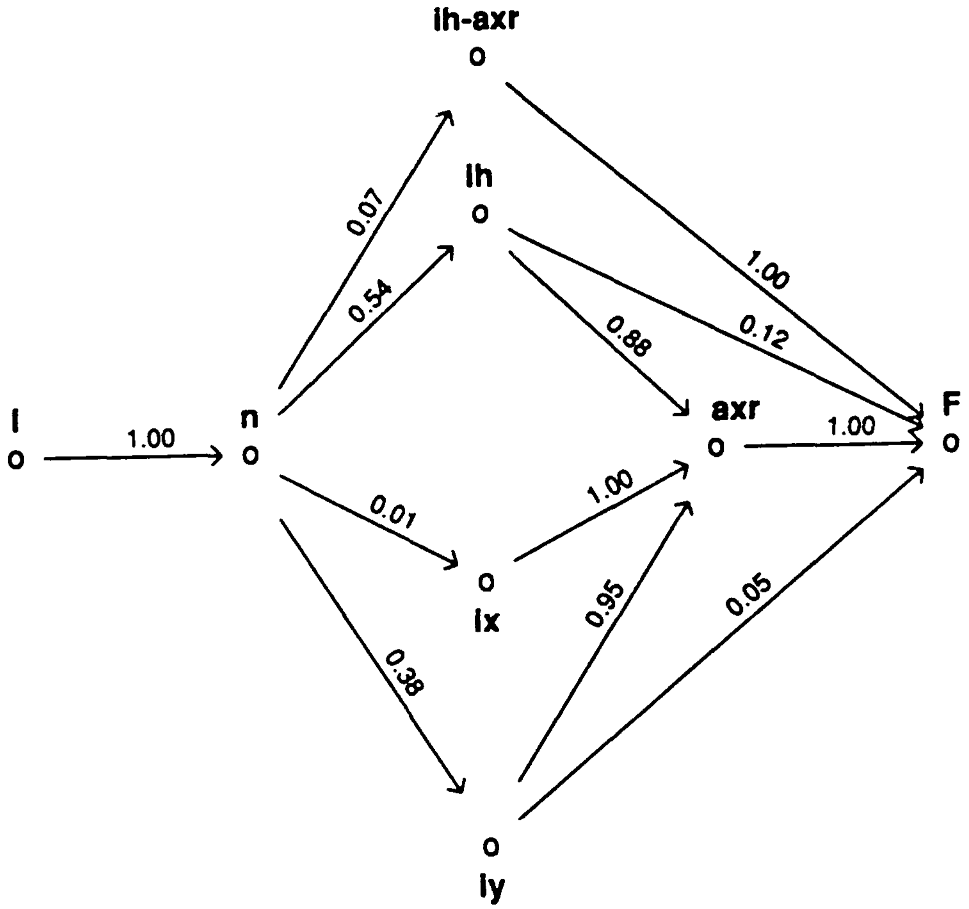


Figure 5.9: Full network for "near."

where q_{CA} is the transition probability between nodes C and A in the original network, $\hat{q}_{CA}$ is the transition probability between the same two nodes in the full network, and $\hat{q}_{C,A\text{-}B}$ is the transition probability between nodes C and A–B in the full network. For the illustrated example, the respective probabilities are .71, .55 and .16. The quantities $\hat{q}_{CA}$ and $\hat{q}_{C,A\text{-}B}$ are determined from the tendency of the segmenter to merge the phone pair AB into a single segment sequence as estimated from the subword model training set. Specifically,

$$\frac{\hat{q}_{C,A\text{-}B}}{q_{CA}} = \frac{\#\text{ segment sequences merged into biphone } A\text{-}B}{\#\text{ occurrences of phone pair } AB}. \qquad (5.3)$$

This equation can be combined with Equation 5.2 to determine $\hat{q}_{CA}$.

Note that this method does not model the dependence of the tendency for the segmenter to merge a phone pair on the keyword in which the phone pair occurs. We elected to ignore this dependency because there may not have been sufficient instances of each keyword in the training set to make good estimates of the transition probabilities if the dependency was taken into account.

Once the pronunciation network was determined, the subword-trained models were built by connecting final and initial states of the subword HMM's in the order specified by the network. Word-trained models with the same topology were also built with the segmental K-MEANS algorithm described above using the subword-trained models as seeds.

Skip Networks

Full networks are much "bushier" than single-pronunciation networks, i.e., they allow many more pronunciations and have many more states and allowable transitions between models. The bushiness poses several potential problems for the word spotter. First of all, the networks might be too unconstrained and thus lead to false alarms. Second of all, there is likely to be insufficient data for training word-trained full networks. Finally, more complex models require more computation in a recognizer's search algorithm.

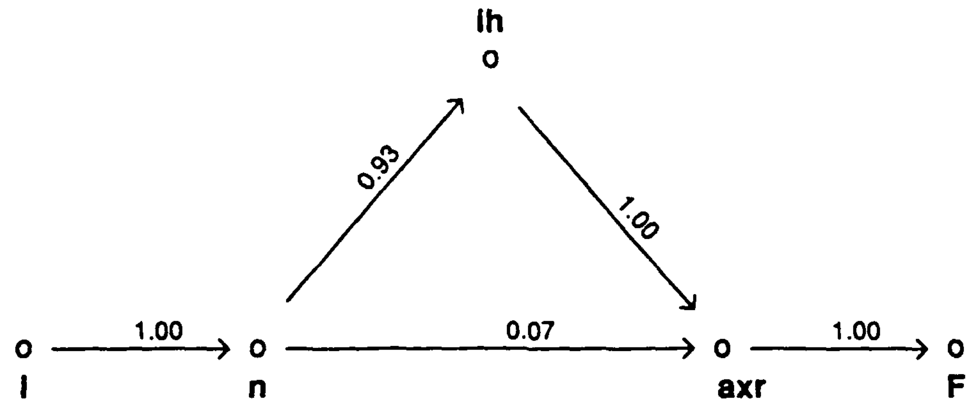


Figure 5.10: Skip network for "near."

Aside from system performance issues, there is ambiguity in determining the cause of performance differences between single-pronunciation and full networks. Differences in performance may be due to the inclusion of biphone models and/or a wider variety of phone models, or due merely to relaxing the requirement on the minimum number of segments required to match the model.

For these reasons, we experimented with skip networks, which are intermediate between single-pronunciation and full networks. An example of such a network for the keyword "near" is illustrated in Figure 5.10.

Each skip network was designed manually to be nearly as simple as the corresponding single pronunciation network but to allow a wider variety of pronunciations. In particular, each skip network tends to allow a minimum number of segments that is much closer to that allowed by the corresponding full network. The skip network deals with segmentation variability by treating a merging of two phones into a single segment sequence as a deletion of one of the phones. For example, the possible merging of phones /ɪɚ/ in "near" is accounted for by the biphone model **ih-axr** in the full network for "near." In the skip network, it is accounted for by allowing the **ih** model to be skipped so that the **axr** model matches the merged phone pair. The skip probability was set in this case to be the same as the transition probability from model **n** to model **ih-axr** in the full network. We could have also accounted for this by allowing the **axr** model to be skipped so that the **ih** model matched the merged pair of phones. However, we judged that this would be a much worse match since it would not account for the retroflexion of the vowel in "near." Judgments of this type were made in building all the skip models. No skip network was built for "from" because there were no biphone models in its full network. Thus, the skip network would have been no different from the single-pronunciation network.

For most words, the skip network was composed out of the same phone models used in the single-pronunciation network, although for the words "can", "nearest" and "what", extra phone models are included to account for perceived deficiencies in the single pronunciation network. Table 5.3 tabulates the differences between the models used in single-pronunciation and skip networks for these three words. The complete set of skip networks is illustrated in Figure 5.11. For the sake of clarity, transition probabilities are not shown in the figure.

	Single pronunciation	Skip network
can	**k ix nx**	**kcl k ix nx**
nearest	**n ih axr ix s**	**n ih axr ix s tcl t**
what	**w ah tcl**	**w ah tcl t**

Table 5.3: Differences in phone models used in single-pronunciation and skip networks. For all other words, the same models are used in both types of network.

Table 5.4 compares the three types of networks in terms of complexity and restrictiveness. In the table, N_T is the number of training tokens available for training the model. "Models" refers to the number of subword models used in

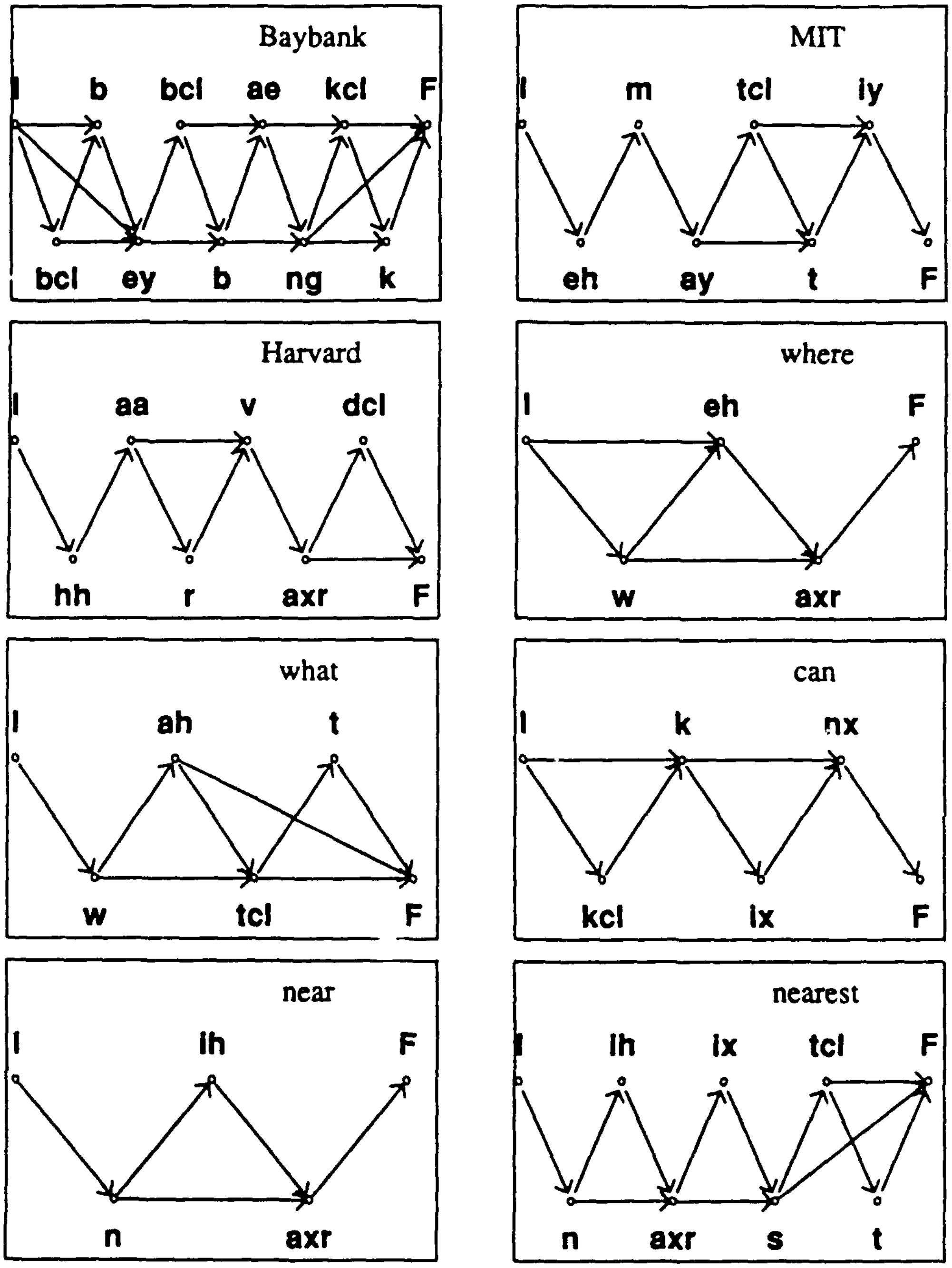


Figure 5.11: Skip networks. Transition probabilities omitted for the sake of clarity.

the network. "States" refers to the number of states for which segment PDF parameters are stored (i.e., not including states with outgoing null transitions only). This is an important measure of model complexity because the word spotting algorithm must compute an acoustic score between each observed segment and each such state. Also, this is the number of states for which acoustic segment data is required for word-trained models. Thus, the higher this number is relative to N_T, the more likely the word-trained model will be undertrained. "Transitions" refers to the number of both null and emitting transitions in the model. This is related to the computation required for finding a path through the network during spotting. Finally, "Min. Segs." refers to the minimum number of segments allowed to match the model. From the table it can be seen that, as stated above, the skip network's complexity is much closer to the single-pronunciation network's than to the full network's.

Results

Table 5.5 summarizes the spotter performance (measured as area over the ROC) observed for the set of keywords as a function of the keyword network type and training mode. The best performance obtained for each word is displayed in boldface. Because of the small number of speakers in the test set and the fact that there are few tokens for some of the words, we doubt that many of the results in the table can be shown to be statistically significant. However, an informal analysis reveals some trends.

The most consistent behavior among all words is the superiority of skip and full networks to single-pronunciation networks. As a rough measure of the superiority of skip networks to single-pronunciation ones, one can consider the fraction of cases in which converting from single-pronunciation to skip networks improves performance when the training mode remains constant. There are 16 such cases tabulated and in 12 of these the skip network performs better. For the same comparison between single-pronunciation and full networks, out of 18 cases, in 15 the full network performs better. Additionally, for the words

Word	N_T	Network	Models	States	Transitions	Min. Segs.
Harvard	88	Single	6	21	48	6
		Skip	6	21	51	4
		Full	18	55	148	5
MIT	67	Single	6	21	48	6
		Skip	6	21	50	5
		Full	8	27	64	5
Baybank	26	Single	9	26	63	9
		Skip	9	26	75	3
		Full	16	53	147	3
from	207	Single	3	12	27	3
		Skip	NA	NA	NA	NA
		Full	10	32	84	2
where	88	Single	3	9	23	3
		Skip	3	9	25	2
		Full	6	21	60	1
wh[illegible]	5[illegible]	Single	3	9	23	3
		Skip	4	12	35	2
		Full	11	35	113	1
can	77	Single	3	7	19	3
		Skip	4	10	28	2
		Full	12	36	105	2
near	48	Single	3	9	23	3
		Skip	3	9	24	2
		Full	6	18	52	2
nearest	44	Single	5	15	37	5
		Skip	7	21	57	3
		Full	18	55	164	3

Table 5.4: Comparison of keyword model types. N_T is number of training tokens for word. Other columns defined in text.

			Area over ROC (%)		
Keyword	N	Training	Single	Skip	Full
Harvard	20	Subword	14.3	5.8	6.2
		Word	6.3	7.7	**4.5**
MIT	26	Subword	17.8	4.3	**4.2**
		Word	34.8	4.7	23.9
Baybank	14	Subword	30.9	**2.8**	14.9
		Word	18.2	6.3	25.0
from	60	Subword	8.1	NA	5.6
		Word	4.2	NA	**2.4**
where	60	Subword	8.3	7.0	**3.6**
		Word	8.2	4.9	5.8
what	56	Subword	47.4	7.1	**4.1**
		Word	29.9	7.4	9.6
can	20	Subword	44.4	25.2	17.6
		Word	20.8	**6.5**	9.6
nearest	17	Subword	7.5	2.0	0.6
		Word	**0.2**	4.5	2.5
near	29	Subword	8.3	12.5	6.8
		Word	**0.8**	0.9	1.3

Table 5.5: Performance by network and training mode. N is number of keyword tokens in test data. Best result for each keyword in boldface.

"MIT", "Baybank", "what" and "can", for which single-pronunciation networks perform particularly poorly using both phone-trained and word-trained models, there are large improvements in performance when skip networks are used, and in seven out of eight cases (the exception being the word-trained model of "Baybank"), performance is much better for the full network than for the single-pronunciation network.

The relative performance of skip and full networks seems to depend the training mode. For example, for the eight keywords for which both skip and full networks were used, in six cases the word-trained skip networks outperformed the word-trained full networks. Additionally, in the two cases where the opposite was true ("nearest" and "where") the advantage of the full networks was small while for the words "Baybank" and "MIT", full networks' performance was much worse. This result might be due to undertraining of the word-trained full networks, especially in the case of "Baybank", which has very little training data available to it and for which the full network is particularly complex. However, such an explanation does not hold for the "MIT" case. Also, one might expect "nearest" to follow the same trend because it to has a very complex full network and not much training but this does not seem to be the case. Thus, while this is a plausible explanation, it is not supported by the data.

The relative performance of skip and full networks is reversed in the subword-trained case. Out of the eight keywords with both types of networks, in only two does the skip network perform better. Because there is more training for the subword-trained models, this may indicate that given lots of training data, a more complicated network is better.

Note that, unlike the single-pronunciation case, word-trained models no longer hold a clear advantage over subword-trained models for the other two networks. This might be because the word-trained models were less adversely affected by the use of single-pronunciation models. The states in each of these models were not forced to be associated with particular subwords and thus could better model the variability in pronunciation of the word. Conversely, the path through a subword-trained model must pass through each subword regardless of how the word is actually pronounced. We show in the next section how this can cause poor performance.

Because of these interactions and the relatively small amount of data, it is difficult to determine a single best combination of training mode and network type from these results, especially one that would generalize to other work in speech recognition. However, for these results, at least, the word-trained skip network is distinguished by having a maximum area over the ROC of 7.7% while the next lowest maximum over all configurations is 17.6% achieve by the subword-trained full network. Thus, this seems to be a good choice for avoiding extremely poor results.

We will return to the comparisons in Section 5.7, where we repeat these experiments with a more effective measurement set. In the next section, we examine why the single-pronunciation network performs so badly for a particular keyword.

5.6 A Case Study of Single- vs. Multiple-Pronunciation Networks

To develop a better understanding of the advantages that skip and full networks hold over single-pronunciation ones, we analyzed the behavior of the "MIT" spotter in greater detail as a function of the network used. In this section, we present a case study comparing the behavior of single-pronunciation and full networks for subword-trained models. The area over the ROC for the former was 17.8% and for the latter was 4.3% so this word was a good candidate for the case study.

The methodology used was to identify trial intervals for which the full network outperformed the single-pronunciation network. It is possible to make a paired comparison of trial intervals because the intervals are the same regardless of the keyword model used in the spotter. To compare spotters for a given keyword trial interval, we determined for each spotter the number of garbage trials which attained a greater score than the score for that keyword trial. It can be shown that the area over the ROC for a given spotter is directly related to the sum of this quantity over all keyword trials. Thus, keyword trials for which this number is high contribute the most to the area over the ROC. Letting this number for keyword trial j be s_j for the single pronunciation network and f_j for the full network, trials for which s_j greatly exceeded f_j were deemed to be candidates for further examination since they were most responsible for the high area over the ROC of the single-pronunciation network compared to the full network. While the same type of analysis could have been applied to garbage trials, we confined our analysis to keyword trials since there were fewer of them.

5.6.1 The Segment Score Plot

For each of these trials, spotter behavior was examined in detail using a data analytic tool we term the *segment score plot.* The plot is similar to tools used by the designers of the SUMMIT system at MIT's Spoken Language Systems Group. Figure 5.12 illustrates the segment score plot for a keyword trial of the "MIT" spotter. The keyword model used was a single-pronunciation subword-trained model. The plot consists of six parts which are time-aligned: (a) a spectrogram, (b) a phonetic transcription, (c) the highest-scoring path through the keyword/garbage network, (d) the highest-scoring path through the keyword model over all paths that span the keyword endpoints determined by the spotter, (e) for the paths in (c) and (d), the segment acoustic scores on the keyword model relative those on the keyword/garbage network, expressed as log probabilities, and (f), an orthographic transcription.[4] We discuss how the scores and state sequences are determined below.

[4] In this example, the automatically determined orthographic transcription seems to be slightly inaccurate since "to" subsumes the /ɛ/ of "MIT". However, this is irrelevant to the

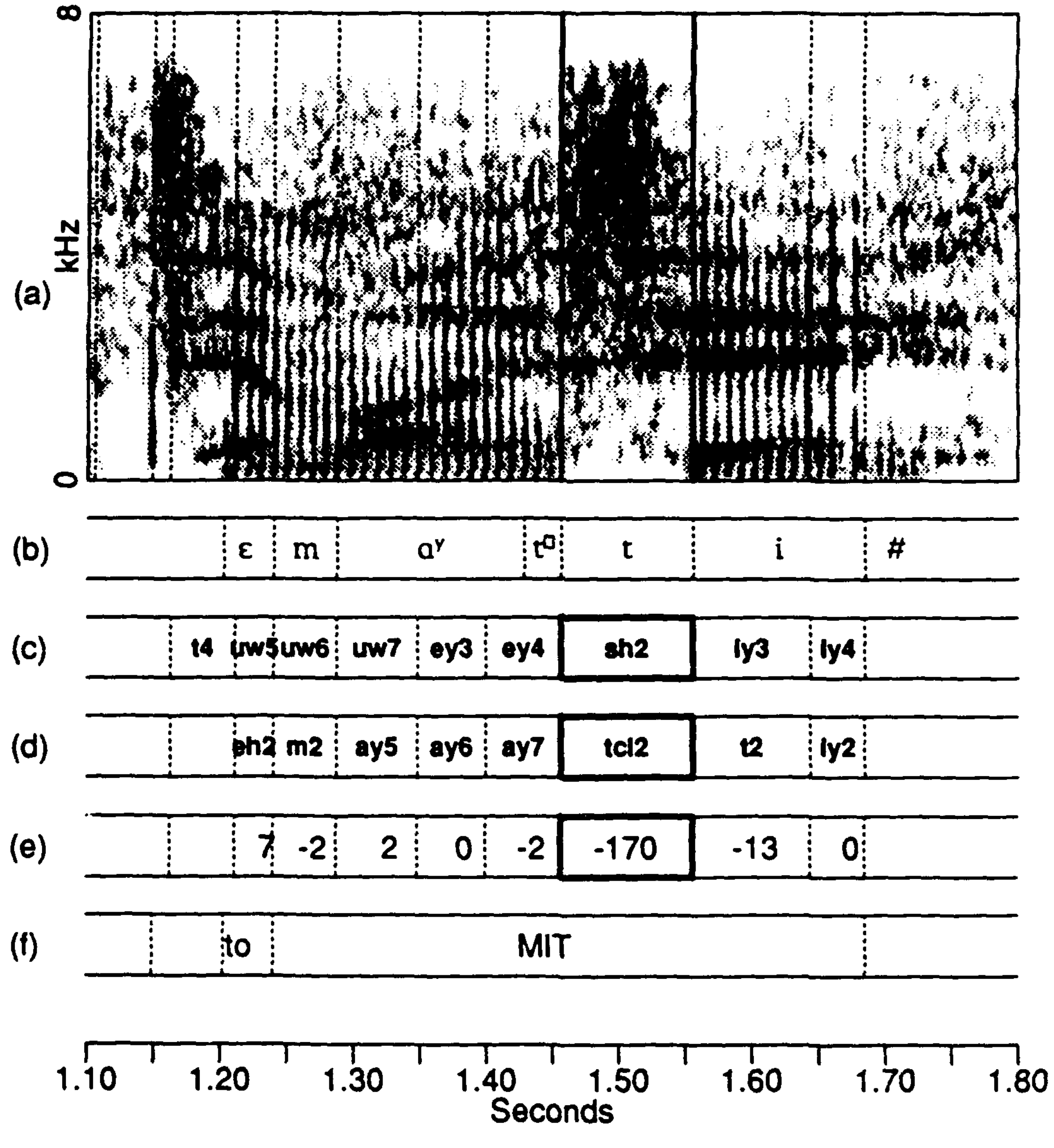


Figure 5.12: The segment score plot. (a) Spectrogram. (b) Phonetic transcription. (c) State sequence through keyword/garbage network. (d) State sequence through keyword model. (e) Relative segment scores. (f) Orthographic transcription. Highlighted segment discussed in text. Times are measured from beginning of utterance.

To interpret the plot, note that highly negative relative scores belong to segments which match much more closely to the keyword/garbage network than to the keyword model. These are the segments that contribute the most to a low score on a keyword trial. Thus, in the example, the highlighted segment whose score is -170 is by far the worst-scoring segment. The labels of the states in the keyword model that are aligned with the keyword trial segments by the Viterbi algorithm appear just above the segment scores. Each state is labelled with its associated subword model and its index in that model. The label **tcl2**, for example, indicates that the state is the second state of the **tcl** subword model that is used to build the model for "MIT." Moving upwards in the figure, the state sequence through the keyword/garbage network appears next. For this example, the state **sh2** is aligned with the poor-scoring segment. We return to this example in the discussion of the case study below.

The segment score plot spans the sequence of segments within the keyword endpoints determined by the spotter scoring algorithm introduced in Section 5.2.1. The path through the keyword/garbage network associated with this sequence is computed by first running the Viterbi algorithm to determine the state sequence that matches the entire utterance and then using the appropriate subsequence. The path through the keyword model is determined by constraining the path to span the endpoints and then using the Viterbi algorithm to determine the best path through the model given the constraints. The constraints are enforced by associating the initial and final states of the model with the initial and final segments of the sequence. Once the paths are determined, acoustic scores λ_{W_t} and λ_{G_t} are computed for each segment t where λ_{W_t} is the score of segment t on the keyword model state aligned with t and λ_{G_t} is the score for keyword/garbage network state aligned with t. Each score is the log of the state PDF measured at the segment's observation vector and is defined in Section 4.8.1 by Equation 4.12. For each segment t, $\lambda_{W_t} - \lambda_{G_t}$ is plotted as shown in part (e) of the figure.

discussion.

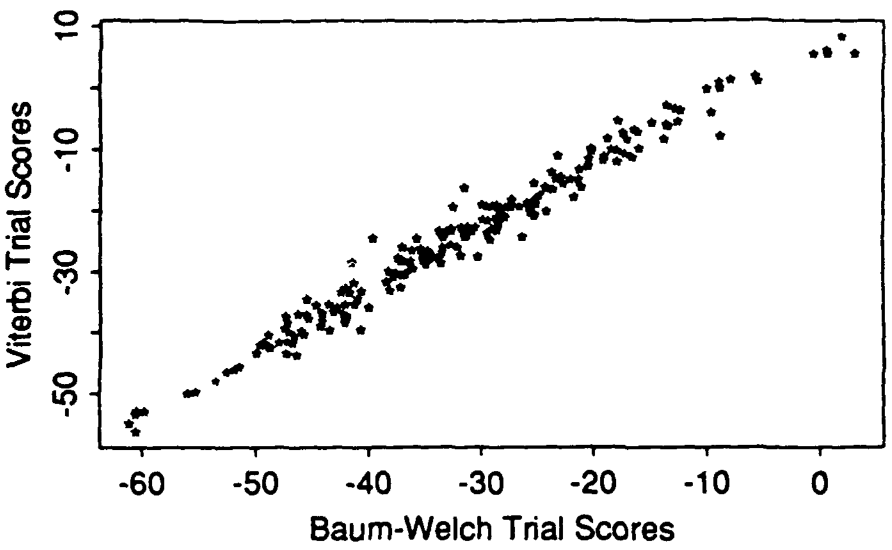


Figure 5.13: Comparison of Viterbi and Baum-Welch trial scores. Sample of 200 trials drawn randomly from an experiment with the "MIT" spotter.

Note that the relative segment scores determined in this manner are not exactly the contributions made by each segment to the trial score determined by the spotter. This is because the Baum-Welch scoring algorithm, which sums scores over all possible paths, is used in the spotter, while the Viterbi algorithm, which computes scores for the best path only, is used for the segment score plot. However, the advantage of the Viterbi algorithm is that it can recover the state sequence associated with the best paths through the keyword and keyword/garbage networks. Because subword labels can be assigned to each state, the plotted state sequences are useful for making a detailed analysis of spotter behavior. Also, the two scoring methods produce similar scores as shown in Figure 5.13, which displays a plot of a random sample of 200 trial scores against those that would be obtained if the Viterbi scoring were used instead of the Baum-Welch scoring. The sample was drawn from an experiment with the "MIT" spotter. There is clearly a close relationship between the two.

While in the present study, the segment score plot is implemented in the context of word spotting, it can be also be used in the continuous speech recognition problem as well. In the general case, recognizer confusions can be studied by comparing the best path through the model of the word actually uttered with that of the state sequence hypothesized by the recognizer.

5.6.2 Case Study Findings

Out of 26 tokens of "MIT" in the test set, we identified seven keyword trials that were ranked much higher by the full subword-trained network than by the subword-trained single-pronunciation network. Segment score plots were produced for both networks for each of these trials and the two plots were compared for each trial. Figure 5.14 displays one such pair of segment score plots. The top figure is a copy of the one illustrated in Figure 5.12 and, as its title indicates, was produced for the single-pronunciation network. The bottom figure pertains to the same trial as scored by the full network. As the titles indicate, for the single-pronunciation case, the keyword trial's score was exceed by almost all (698 out of 711) of the garbage trial scores while in the full network it was exceeded by a small fraction (12 out of 711) of them.

It is clear from the top plot that the low score is caused primarily by the segment which is highlighted, since the segment relative score of -170 is by far the most negative score observed for any segment. The low score is due to the fact that the single-pronunciation network constrains the state sequence to include at least one state of the **tcl** model. This model is supposed to match /t/ closures but from the spectrogram it appears to be aligned with the /t/-aspiration. In fact, for this token, the closure is poorly articulated since the region of the spectrogram aligned with the closure transcription label has a large amount of energy. Note, too, that the model for the /t/ burst and aspiration must follow the model for the closure in the pronunciation and thus is matched to the first part of the /i/, leading to a segment score of -13.

The same segment in the bottom plot is matched by the biphone model

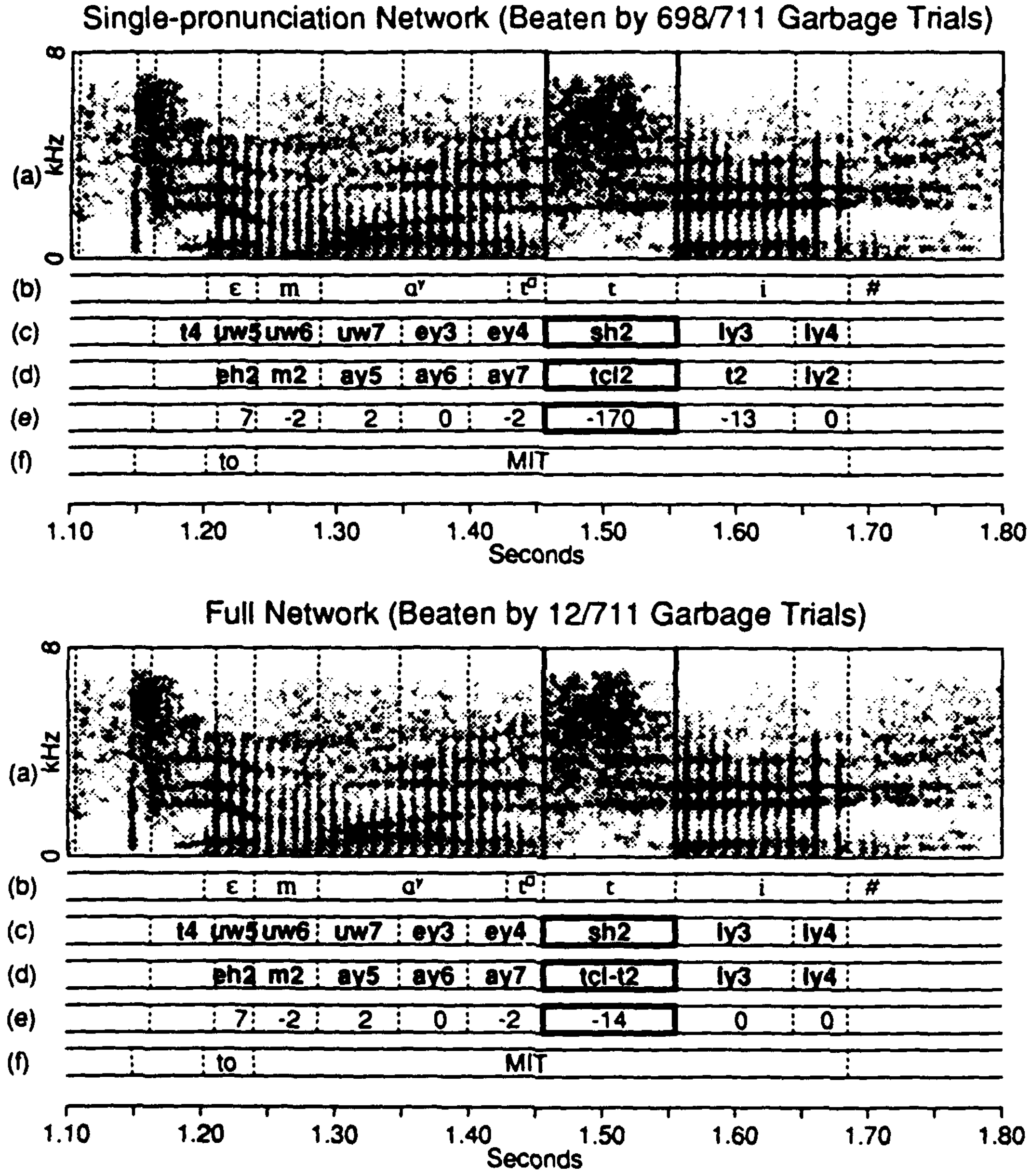


Figure 5.14: Segment score comparison of single-pronunciation and full networks for "MIT" for a trial in which the full network performs better. (a) Spectrogram. (b) Phonetic transcription. (c) State sequence through garbage-keyword network. (d) State sequence through keyword model. (e) Relative segment scores. (f) Orthographic transcription. Highlighted segment discussed in text. Times are measured from beginning of utterance.

tcl-t. The segment also attains a fairly low score of -14, perhaps because the aspiration is very long in duration and so might match an unvoiced fricative like /š/ better, as indicated in part (c) of the plot. However, the match is much better than that attained for the closure and it allows the **iy** model to match the /i/, so as to achieve a score of 0 on those segments. While we show no examples here, the skip network was also able to deal with the poorly articulated closure, by skipping over the **tcl** model.

For each of the seven tokens investigated with this method, the same phenomenon occurred. There was always a poor closure and always a very low score on the segment matched to the **tcl** model. Also, the seven tokens were divided among two of the five speakers in the test set, indicating that the phenomenon is not just an idiosyncrasy of a particular speaker.

To verify that there was not some other cause for the poor performance of the single-pronunciation network, we produced segment score plots for keyword trials that received high scores from *both* of the two networks. An example is illustrated in Figure 5.15. Note that the closure (whose segment is highlighted) is well-articulated. The state labels in the keyword/garbage network path (part (c) of the figure) that consist solely of numbers refer to states in the keyword model. Thus, for both networks, the best path through the keyword/garbage network includes the keyword model. Because the best state sequences through both the keyword model and keyword/garbage network are the same, the relative acoustic score on each segment is 0, as shown. This example is typical of the keyword trials which received high scores from both spotters. In all such cases, the closure was well-articulated, indicating that the chief difference between the two types of network was in their ability to model the poorly-articulated closure.

One point worth noting is that the high-scoring keyword tokens with well-articulated closures were mainly associated with different speakers than the tokens with poorly-articulated closures. This is further evidence of the potential ability of speaker-adaptation schemes to achieve improved recognizer

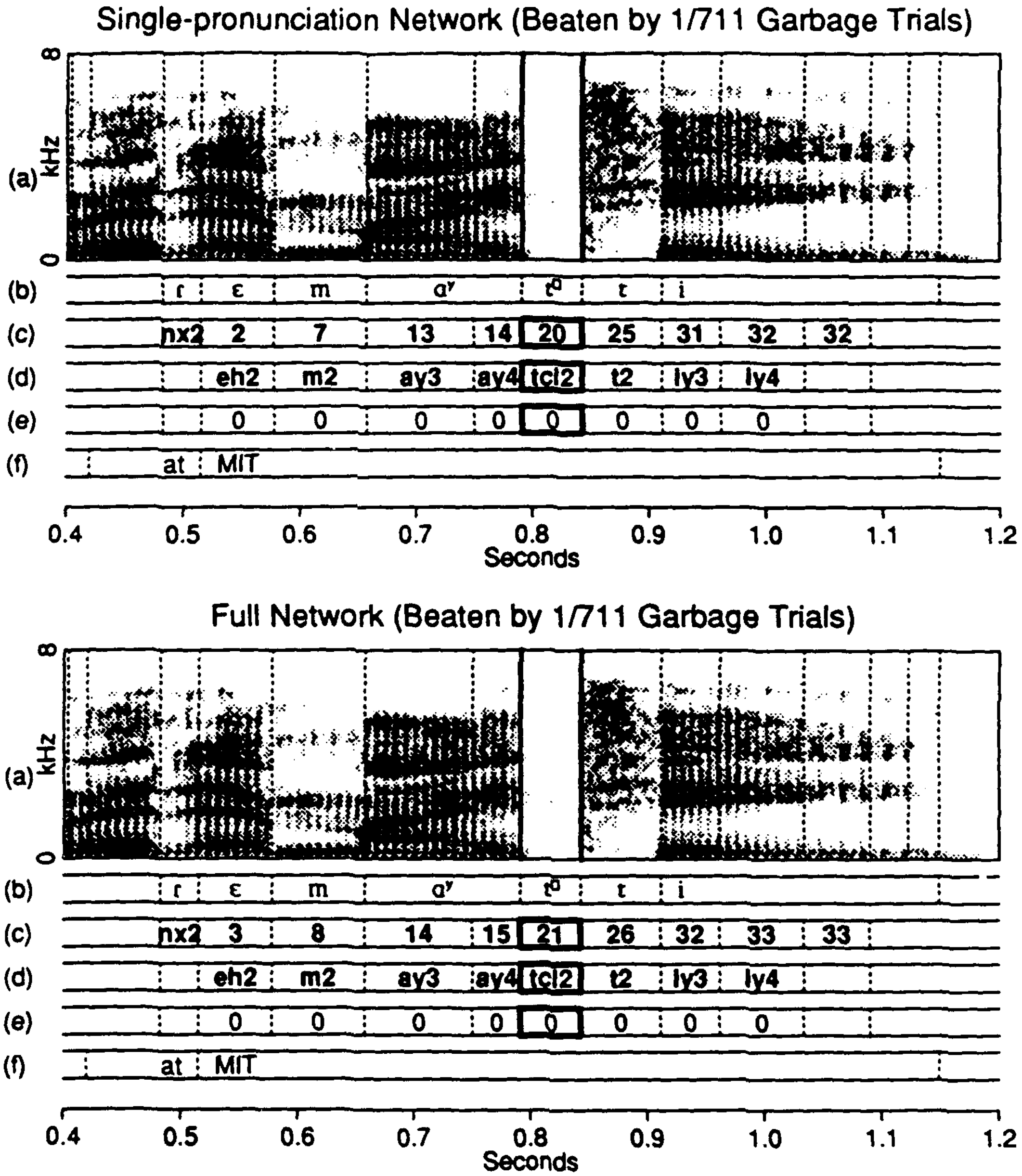


Figure 5.15: Segment score comparison of single-pronunciation and full networks for "MIT" for a trial in which both networks perform well. (a) Spectrogram. (b) Phonetic transcription. (c) State sequence through garbage-keyword network. (d) State sequence through keyword model. (e) Relative segment scores. (f) Orthographic transcription. Highlighted segment discussed in text. Times are measured from beginning of utterance.

performance my taking advantage of the tendencies of speakers to exhibit consistency from token to token.

5.6.3 Discussion

The case study findings have implications that are specific to the case itself as well as some that are more general. The most specific interpretation of the case study is that in the word "MIT", the /t/ closure may be poorly articulated and if this is not accounted for in the model for the word, spotter performance will be poor. The possibility that a closure may be poorly realized is well-known by phoneticians. It has been dealt with in the context of speech recognition by Lee [Lee 88], for example, who built a single HMM for /t/ that is flexible enough to allow the closure to be skipped rather then by devoting specific models to closures and burst-aspirations. Thus, the finding itself is not new.

However, the results of the study indicate the extent to which the failure to account for phenomena such as this one can affect recognition results. For the word "MIT", the spotter based on the single-pronunciation keyword model produced poor scores on almost 1/3 of the tokens solely due to a single phenomenon. While we have not subject the other keywords in the present study to the same analysis, it is likely the case that the poor performance of single-pronunciation networks in these cases might be related to a small number of phenomena as well.

The fact that the segment score plot was successful in identifying the unmodelled source of variability and could presumably be used to do so in general is more important than the results of this particular case study. For the problem of modelling pronunciation variability, the plot can be used to identify places where a word model is too strict in that it does not account for variability. Conversely, it can be used to show that only a limited number of pronunciations must be accounted for in the model, so that underconstrained models are not employed. This is important because the results of [Weintraub 89] that were discussed above and those of the present work indicate that models with too little constraint can degrade performance just as models with too much constraint can.

Tools such as the segment score plot can be incorporated in an iterative process of building pronunciation models. In the first step of the process, models are built based on prior knowledge. Then errors are analyzed and the models refined and the process is iterated. By looking for patterns in the data, possible model refinements might be suggested that would not be considered if purely automated techniques were used in model building. In this case study, for example, the fact that the quality of the closure articulation appears to be highly speaker-dependent might suggest further study that could lead to improved procedures for speaker adaptation. We discuss this process of model building in greater detail in Chapter 6.

5.7 The Effect of Measurement Set on Performance

We repeated several of the experiments described in the previous section using the 39-discriminant measurement set defined in Section 4.7.3. As reported in Chapter 4, substantially better phonetic recognition results were achieved using this set than obtained with the baseline set used in the experiments of the previous section. The main goal of the experiments reported here is to compare the two measurement sets in a task more closely akin to word recognition to see if the 39-discriminant measurement set maintained its advantage. Such a finding would suggest that the results of Chapter 4 can be generalized to a more complex task such as word recognition.

A secondary goal of the experiments is to test if there are any interactions between the measurement set used and the relative performance of the various types of keyword model discussed in the previous section. However, we did not repeat the experiments involving the single-pronunciation networks because their overall poor performance in the previously reported experiments made

	Area over ROC (%)		
	Baseline	39 Discs.	N
Harvard	7.7	0.6	20
MIT	4.7	0.4	26
Baybank	6.3	0.6	14
from	2.4	1.0	60
where	4.9	2.9	60
what	7.4	4.1	56
can	6.5	2.6	20
nearest	4.5	0.1	17
near	0.9	1.9	29

Table 5.6: Effect of measurement set on word spotting performance. All results for word-trained skip networks except for those for the word "from", for which word-trained full networks were used. N is number of keyword test tokens.

them unworthy of further study.

Thus, the results of these experiments along with those of the previous section involve four factors: the keyword, the training mode (subword- or word-trained), the network type (skip or full) and the measurement set (baseline or 39-discriminant). We first report results comparing the two measurement sets on the word-trained skip network, which was judged to be the best overall in the previous section. For "from", we use the full network since no skip network was built for that word. The results are tabulated in Table 5.6.

The results indicate that the measurement set improves performance in all cases but one. The effect seems to be more pronounced for content words than function words and in fact the results for the 39-discriminant set are more intuitively reasonable than those of the baseline set in that function words now have higher areas over the ROC than content words. This is to be expected because they are harder to distinguish from other speech.

The 39-discriminant results for the word-trained skip network are the best achieved overall for all combinations of training method, network type, and measurement set we have considered. The superiority of this factor combina-

			Area over ROC (%)			
			39 Discs.		Baseline	
Keyword	N	Training	Skip	Full	Skip	Full
Harvard	20	Subword	1.9	2.8	5.8	6.2
		Word	**0.6**	1.8	7.7	4.5
MIT	26	Subword	**0.2**	**0.2**	4.3	4.2
		Word	0.4	0.5	4.7	23.9
Baybank	14	Subword	2.1	2.0	2.8	14.9
		Word	**0.6**	1.4	6.3	25.0
from	60	Subword	NA	1.5	NA	5.6
		Word	NA	**1.0**	NA	2.4
where	60	Subword	6.1	**2.0**	7.0	3.6
		Word	2.9	3.8	4.9	5.8
what	56	Subword	5.1	4.4	7.1	**4.1**
		Word	**4.1**	4.5	7.4	9.6
can	20	Subword	12.2	5.4	25.2	17.6
		Word	**2.6**	3.0	6.5	9.6
nearest	17	Subword	7.8	0.3	2.0	0.6
		Word	**0.1**	0.2	4.5	2.5
near	29	Subword	29.6	10.6	12.5	6.8
		Word	1.9	1.9	**0.9**	1.3

Table 5.7: Complete spotting results. Best performing configurations for each word in boldface (including ties to one decimal place). N is number of keyword test tokens.

tion can be seen in Table 5.7, which summarizes all the results of this chapter, except for those obtained with single-pronunciation networks. Of all factor combinations, this combination performed best or tied for best for five of the nine words, as can be seen by considering the best result for each word, which is set in boldface. Also, the area over the ROC for the worst-performing spotter - 4.1% on "what" - was smaller for this combination of factors than for any other combination. Thus, the word-trained skip network appears to be the best choice regardless of measurement set.

Inspection of the table also reveals that the 39-discriminant measurement set is superior regardless of the other factor levels. For eight out of the nine words, this measurement set provides the lowest or tied-for-lowest area over the ROC. Also, the 39-discriminant set achieves higher performance in 28 of the 34 comparisons where all factors are fixed except for the measurement set, The word "near" is an anomaly in that for all four combinations of training mode and network type the baseline set performs better. We have not pursued an explanation for this result. Another striking result is the huge effect the measurement set has on the performance achieved for the words "MIT" and "Baybank" when word-trained full networks are used. Again, this matter has not been pursued. It may be due to an artifact of the relatively small training and test sets available for these words.

The table also reveals that the same interaction between training mode and network type holds for both measurement sets: subword-trained models employing full networks outperform those employing skip networks. We discussed the reasons for this in Section 5.5.4.

However, one difference between the results on the baseline and the 39-discriminant measurement sets is that word-trained models seem to perform better with the latter measurement set regardless of network type and regardless of whether the keyword is a function or content word. For seven out of eight words, the word-trained skip network outperforms the subword-trained skip network and for six out of nine words, the word-trained full network outperforms the skip-trained one. For the words "can" and "near", there are instances where the word-trained network has a particularly large advantage. Recall that for the baseline measurement set, the word-trained models only outperformed subword-trained models for the skip network. One plausible explanation for the apparent interaction between measurement set and training mode is that the 39-discriminant set on a given segment is based on measurements made well beyond the segment while the only out-of-segment measurements made in the baseline set are made 5 ms after the segment's end. For a segment associated with a particular subword, the out-of-segment measurements are particularly dependent on the identity of the phones surrounding the subword. For example, if a vowel follows a stop consonant, the spectrum 35 ms previous to the start of the first vowel segment is likely to be more influenced by the stop's identity than the vowel's. Thus, for the word "MIT", for example, the spectrum 35 ms before the start of the first segment associated with the /i/ will reflect the influence of the /t/. However, the subword-trained model for the /i/ is trained from instances of /i/ appearing in all contexts. Thus, the estimated PDF for the measurements associated with that segment will not be tuned to the specific context in which the /i/ appears in "MIT." However, for the word-trained model, the state in the model that tends to be associated with the /i/ will be trained only from segments for which a /t/ precedes the /i/. Thus, the state parameters will be tuned to the proper context. While the same argument can be made regardless of the measurement set used and is the justification for using context-dependent modelling in the first place, it applies to an even larger extent when measurements are made well beyond the segment boundaries.

The apparent interaction of measurement set and training mode is worth further study. If it is confirmed, then it is likely that segment-based schemes that include measurements made beyond segment boundaries will be particularly able to benefit from modelling context with triphone or word-specific models for example.

5.8 Summary

In this chapter, we extended the segment-based HMM to word modelling. We examined several word modelling issues, relevant both to HMM's in general and segment-based HMM's in particular. The issues we looked at were

1. training mode: i.e., whether a word model was trained from specific instances of the word or by concatenating context-independent subword models,

2. pronunciation network type: i.e, the variability in pronunciation allowed by the network, and

3. acoustic measurement set.

For each word in the study, the investigation was conducted by gauging the performance of a spotter for that word as a function of the word model used. In the course of the investigation, we developed novel algorithms for word spotter scoring and performance evaluation. The algorithms have several advantages over those that have been used previously. In particular, they have the ability to

1. determine both the beginning and ending points of a putative hit,

2. generate a smooth receiver operating characteristic (ROC) in a computationally efficient manner, and

3. compare word spotters on the same task using a non-parametric significance test.

Because the algorithms find both the beginning and ending points of a putative hit and because they divide incoming speech into discrete "trials", they are particularly suited to detailed error analysis, one of the main topics of the thesis.

We found that word-trained models usually outperformed subword-trained ones. The advantage increased with the restrictiveness of the network. This finding is probably due to the fact that there are fewer states in more restrictive pronunciation networks than in less restrictive ones. Thus, the relatively small amount of data available for training word-specific models does not pose as much of a problem for the more restrictive networks. Word-trained models performed better than subword-trained ones even for content words for which there were a relatively small amount of training data. This finding buttresses the well-known fact that models which account for phonetic context usually outperform those that do not.

The networks that performed best overall were word-trained skip networks, which were intermediate in flexibility between single-pronunciation and full networks. Thus, we conclude that while some flexibility in a word model is desirable, it is disadvantageous to use networks that are overly bushy, since for word-trained models, overly bushy networks have too many states to train reliably. We noted, however, that there was an interaction between network topology and training mode so that when subword-trained models were used, full networks performed best. Presumably, this occurred because there was no shortage of training data for subword-trained models. Thus, given enough training data, it might be advantageous to use networks that allow a large variety of possible pronunciations. Single-pronunciation networks performed particularly badly. We attribute their poor performance to their inability to model either segmenter or pronunciation variability.

We introduced a technique for detailed error analysis termed the *segment score plot* and used it to demonstrate that for the word "MIT," the single pronunciation network failed because it was unable to account for variability in the realization of stop closures. We concluded from this case study that detailed error analysis techniques could be useful for discovering model deficiencies.

Finally, we found that the effect of measurement set on performance was similar to that found in the phonetic recognition task of Chapter 4: a measurement set that included adjacent-segment measurements outperformed one that did not by a wide margin for almost all words in the study.

For the best-performing choice of training mode, network type, and measurement set, the relative spotting performance among words in the study met our expectations. Confusable function words such as "can" and "what" were more likely to be confused with other speech than were content words such as "Harvard" and "MIT."

Chapter 6

The Application of Exploratory Data Analysis to Speech Recognizer Development

Exploratory data analysis (EDA) encompasses a variety of techniques for examining a set of data for the purposes of discovering structure in it and for building statistical models to account for the structure. We believe that the philosophy and strategies of EDA have a much larger role to play in the development of speech recognition systems than they have played thus far. In this chapter, we outline the major ideas behind EDA, discuss the role they have played in the work we have presented in previous chapters, extend techniques developed in Chapter 5 for detailed word spotter error analysis and conclude with a discussion of why the time is ripe for further use of EDA in speech recognition development.

The major technical contributions of this chapter are methods for making detailed diagnoses of speech recognizer errors. We use these methods in a case study and show that a small number of phenomena are disproportionately responsible for the errors found in the study. Thus, a small number of improvements in the models would presumably lead to a large performance increase. While we did not attempt to make these improvements, we discuss how this could be done were it within the scope of our work.

6.1 EDA and its Role in Our Work

Rather then provide our own description of EDA, we will quote liberally from John Tukey, its best known proponent, and M. B. Wilk, using excerpts from their paper "Data Analysis and Statistics: An Expository Overview." [Tukey 86].

Tukey and Wilk have this to say about the intent of EDA:

> The basic general intent of data analysis is simply stated: to seek through a body of data for interesting relationships and information and to exhibit the results in such a way as to make them recognizable to the data analyzer and recordable for posterity. Its creative task is to be productively descriptive, with as much attention as possible to previous knowledge, and thus to contribute to the mysterious process called insight.

They also claim "exposure, the effective laying open of the data to display the unanticipated" to be a "major portion of data analysis." For use in model building, the authors emphasize the value of an iterative approach in which a model is fit to the data, the model's residuals (differences between the predicted and observed values of the data) are examined, and information obtained from the investigation is used to build a better model. They consider fitting to be an essential part of data analysis:

> The single most important process of data analysis is fitting. It is helpful in summarizing, exposing and communicating. Each fit (1) gives a summary description, (2) provides a basis for exposure based on the residuals, and (3) may have the parsimony needed for effective communication.

Examination of the residuals is considered a crucial part of the process and many of the techniques of EDA were developed for this purpose.

The means for attaining the goals of EDA are summarized by the authors as follows:

> In addition to the two-pronged use of summarization and exposure, including careful attention to residuals, three of the main strategies of data analysis are:
>
> 1. Graphical presentation.
> 2. Provision of flexibility in viewpoint and facilities.
> 3. Intensive search for parsimony and simplicity, including careful reformation of variables and bending the data to fit simple techniques.

Graphical techniques are particularly important for revealing relationships in the data, especially for large data sets, since humans are believed to be better at finding patterns in plots of many data points than in tables of numbers.

We have used some of these strategies in previous chapters. Our chief intent has been to build better subword and word models for speech recognition, not to find patterns for their own sake or to parsimoniously summarize data for the purposes of communication to others, although in some cases the results are of interest in their own right.

For example, the development of the regression model for F_1 and F_2 described in Section 4.4 included almost all the aspects of EDA cited above. We first fit a linear model to predict the complete range of the formant and noted using a scatter plot that the fit was poor in certain ranges. The pattern of the residuals suggested reducing the range of formants predicted. We did this and improved the fit. Finally, we reasoned, using previous knowledge of the relationship between the formants and the MFSC filter outputs, that the non-linear center of gravity transformation would provide a better fit, particularly if the filter outputs were exponentiated. In the terminology of Tukey and Wilks, this represented a reformation of the variables and a bending of the data to fit simple techniques (in this case multiple regression). The reformation did indeed lead to a better fit.

Other examples of EDA's use in model building are the perspective plots used to determine the locations for making out-of-segment measurements in Section 4.5 and the dendrogram used to display the covariance clustering results in Section 4.7.3. Finally, the plots of class centers for the vowel/semivowel discriminants in that section exemplify the use of EDA for insight and communication.

In each of these cases, general-purpose data analytic techniques were used. In Chapter 5, however, we developed the segment score plot, which is more specialized to speech recognition. This technique fits into the EDA paradigm as well. First, we identified trials for which the word spotter performed poorly. These represent large "residuals" in the sense that the word spotter is a model for predicting an output (either a keyword or garbage) from an input (a sequence of observation vectors). Word spotting errors represent cases in which there is a large difference between the predicted and observed output. The same can be said of errors made by word recognition systems.

After identifying the large residuals, we studied them more closely, using segment score plots to reveal their structure. We recognized poor /t/ closures from the spectrograms of "MIT" and noted that the scores for segments aligned to the **tcl** model were extremely low. Because this pattern occurred in all the low-scoring keyword trials and never in the high-scoring ones, we could explain the poor performance of the single-pronunciation models parsimoniously. As we were not looking for this particular problem, the techniques resulted in "exposure" of the "unanticipated," in the words of Tukey and Wilk.

We should point out that EDA was not used in this case to build a better model. In fact, we had suspected that single pronunciation models might be overly restrictive and so experimented with less restrictive models before the analysis. However, because the single-pronunciation model's inadequacy can be attributed to a single cause, we learned that a better model for "MIT" might be built by modifying the model to allow deletion of the closure but not to allow alternate paths. Thus, as we pointed out in Chapter 5, the segment score plot could be useful for building appropriately restrictive word models.

More generally, we could examine the closure deletions more closely to find patterns relating deletion occurrences and other factors, such as the speaker's identity and the position of "MIT" within the sentence. Patterns found in this way could be used to formulate hypotheses about the relationships between closure deletion and these other factors and the hypotheses could be tested on new data. Thus, EDA could be used as a means for hypothesis formulation, which Good [Good 83] identifies as one of its major aims. If an hypothesis is confirmed in this manner, a model for predicting closures could be built that could be incorporated into the word spotter to improve its performance.

However, it is unlikely that most errors are severe enough that they can be understood with the segment score plot alone. Thus, there is a need to develop techniques for understanding errors in more detail. This is the subject of the next section.

6.2 Specialized Techniques for Speech Recognizer Diagnostics

In the discussion that follows, we will describe each technique and its user interface. The interface is important because the goal of the techniques is to communicate information to the system designer in a form that is easy to interpret. Towards this end, most of the information is displayed graphically. Also, to be worth the time expended, the techniques must be convenient to use. Towards this end, they have been implemented on a workstation and are summoned either by menu or by clicking a mouse at appropriate places on mouse-sensitive displays. The organization is hierarchical so that clicking on each display allows the user to explore recognizer behavior at a deeper level.

We illustrate the techniques developed in this section with a case study which was aimed at understanding one of the findings of Chapter 5. From Table 5.7, it can be seen that while for most words in the study, the word-trained model outperforms the phone-trained one when the 39-discriminant set is used, the effect is most striking for the word "near." We wished to determine why this was so by investigating in detail the performance of the full network word- and subword-trained models. While we ended up focusing on another phenomenon and never answered the original question, we will begin the discussion with our initial approach to answering it.

6.2.1 The Paired Trial Scatter Plot

In a study comparing the performance of two speech recognizers or spotters tested on the same data, it is worthwhile to identify tokens for which the two systems produce different results. This was the strategy we used in Chapter 5 to learn why the single-pronunciation network performed so badly on the word "MIT." To conduct such a study in the context of our word spotter, we use the *paired trial scatter plot.* Figure 6.1 displays an example of such a plot. The plot in the figure compares the word- and subword-trained spotters of "near." As we pointed out in Section 5.6.1, a trial-by-trial comparison of the two spotters is possible because the trial intervals are the same for both spotters. Each trial interval is represented on the plot by a point whose co-ordinates represent the ranks of the scores computed for the trial interval for the two spotters. The rank is defined in ascending order of scores, so that the rank of the lowest score is 1, for example.

In this case, there were 1253 trials in all, including 29 keyword trials. The keyword trial denoted "Text example" was ranked 1022 when word-trained models were used in the spotter and 743 when phone-trained ones were used. This is represented by an "o" at the point (1022,743) on the plot. The line $y = x$ is superimposed on the plot. Thus, keyword trial points that are far below this line represent trials that are ranked much more highly by the word-trained- than by the phone-trained spotter. Since for good spotter perfor-

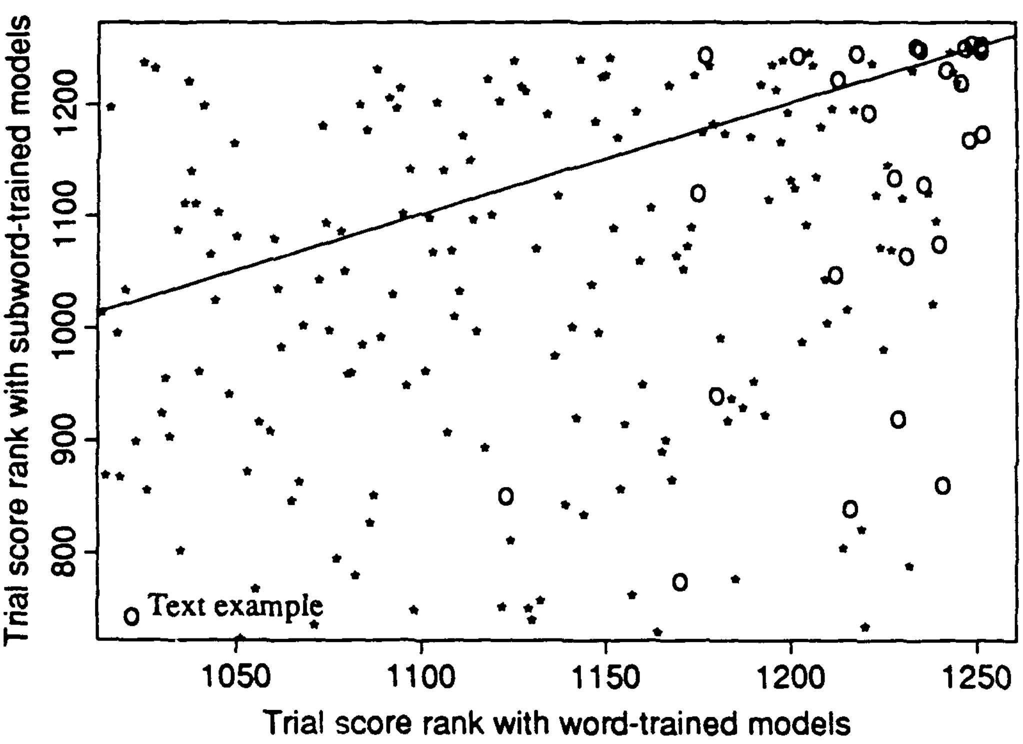


Figure 6.1: Paired trial scatter plot of phone- and word-trained spotters for "near." Trial score ranks for each spotter are computed over all 1253 keyword and garbage trial spotter scores. However, only those trials which score higher than the lowest-ranking keyword trial are plotted. Aster'sks (*) and o's represent garbage and keyword trials, respectively. The line $y = x$ is superimposed on the plot.

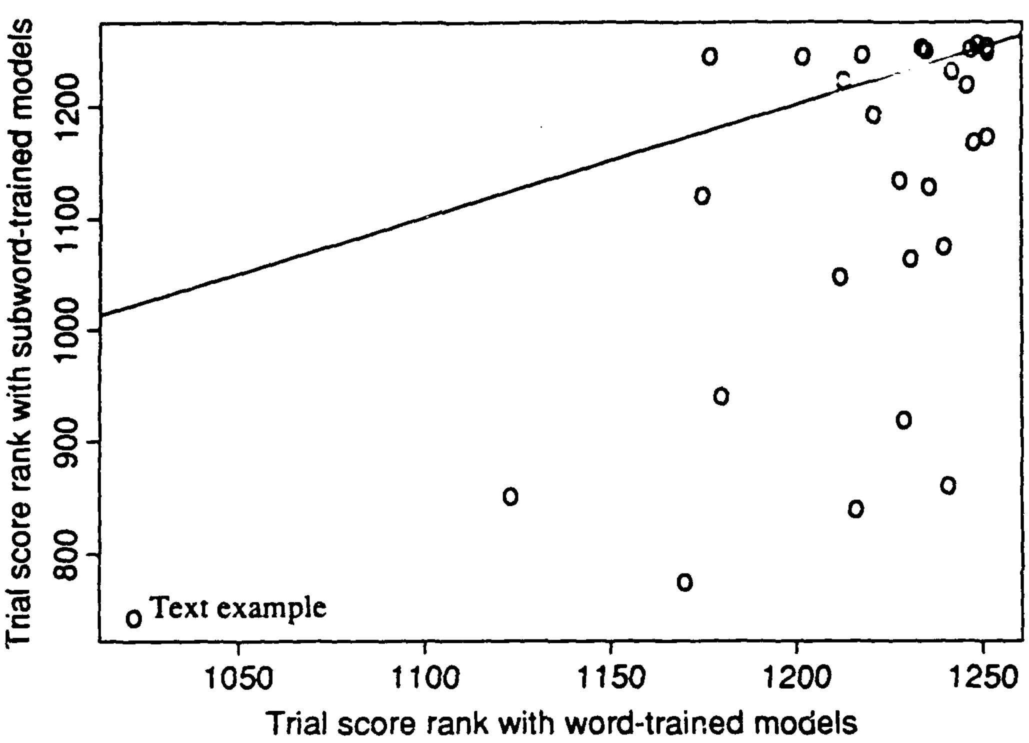


Figure 6.2: Paired trial scatter plot of phone- and word-trained spotters for "near" - keyword trials only. Keyword trial score ranks for each spotter are computed over all 1253 keyword and garbage trial spotter scores. The line $y = x$ is superimposed on the plot.

mance, one wants keyword trials to have a tendency to be ranked higher than garbage trials, these points represent trials for which the word-trained spotter outperforms the phone-trained one. Conversely, for garbage trials, points below the line indicate better performance by the word-trained spotter. Note that most of the keyword trial points are below the line. This indicates that the *word-trained spotter* outperforms the phone-trained one overall.

The plot can be simplified by considering only keyword trials, since there are fewer of these. This is useful if one wants to concentrate on these trials alone, as we did for the study. An example is shown in Figure 6.2. The keyword trial points whose positions are the furthest below this line were the the most promising ones for gaining insight into the superiority of the phone-trained models.

However, we were also curious about trials that were ranked poorly by both spotters. Thus, we clicked on the trial denoted "Text example" and generated the paired segment score plot illustrated in Figure 6.3.

Note that while the keyword being spotted is "near", the word associated with the keyword trial in this case is "nearest", since it includes the morph "near." The thick lines in the plot represent the putative keyword endpoints for each spotter. The most striking aspect of this plot is that in both cases, the best path through the keyword model is misaligned with the orthographic transcription of the keyword. In the word-trained case, the second state of the **n** model is aligned with the $/i^y/$ and the second state of the **ih** model is aligned with the $/i^y/$ -/r/ sequence. For some reason, the word spotter avoided a path through the keyword model that was aligned with the /n/ in "nearest." The phone-trained model also was misaligned in that the best path did not included the /r/ in "near." We obtained similar results for nearby points on the scatter plot. The cause of the misalignments appeared to be a more important problem to study than the superiority of the phone-trained model. Thus, we decided to focus on it. For the remainder of this section, we discuss our investigation of the alignment problem, focusing on keyword tokens that were assigned low scores by the phone-trained spotter. The word-trained spotter exhibited similar behavior on these tokens.

6.2.2 The Trial Score Scatter Plot

To graphically represent individual spotter trials and allow them to be conveniently investigated in more detail, we use a scatter plot which arranges trials according to their scores, as shown in Figure 6.4. As in the paired trial scatter plot, the asterisks (*) and o's represent garbage and keyword trials, respectively. So as to reduce crowdedness, only garbage trial scores that

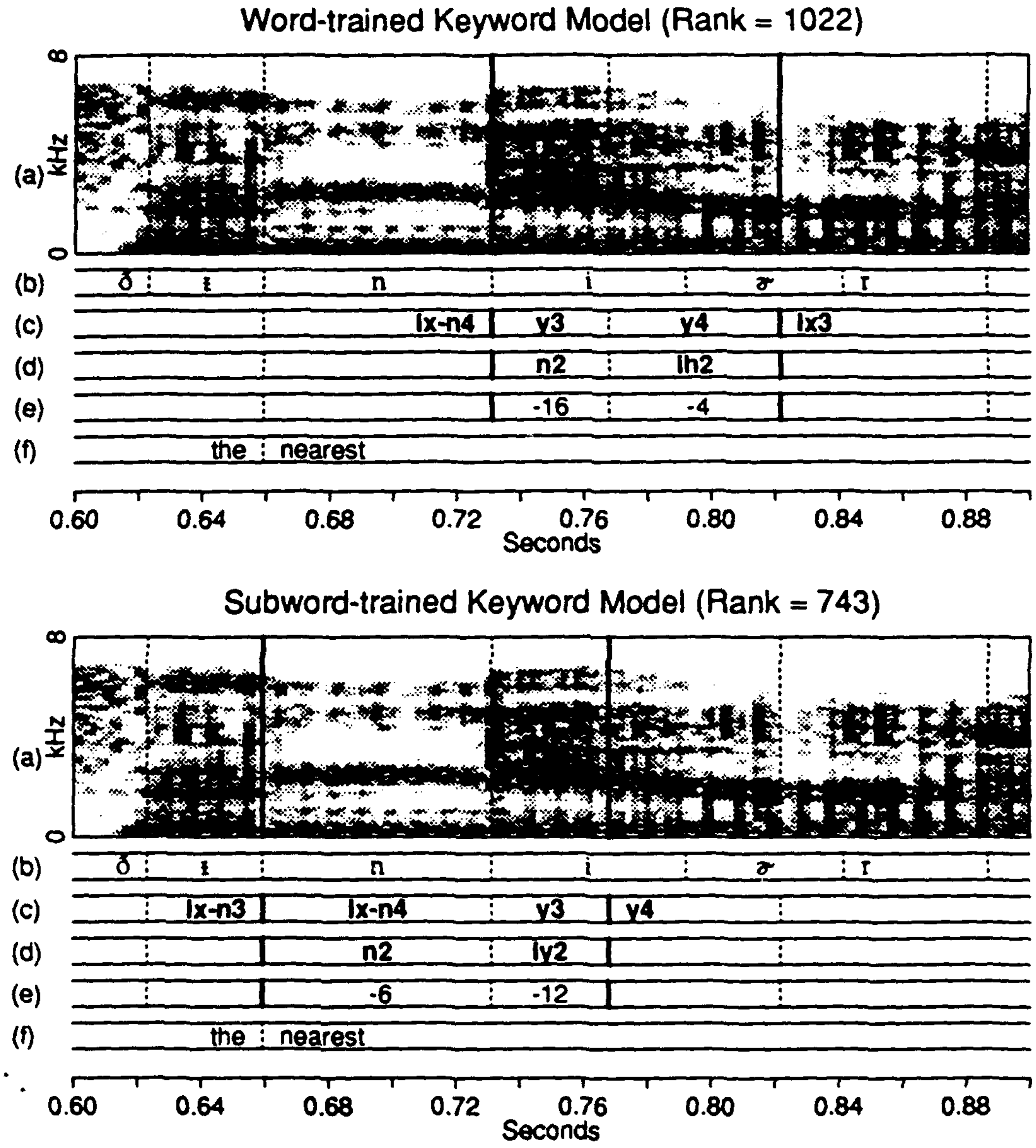


Figure 6.3: Paired word- and subword-trained segment score plots for "near" for trial discussed in text. Top and bottom plots are for word-trained and phone-trained models, respectively. In both cases, the full network keyword model and 39-discriminant measurement set were used. (a) Spectrogram. (b) Phonetic transcription. (c) State sequence through keyword/garbage network. (d) State sequence through keyword model. (e) Relative segment scores. (f) Orthographic transcription. Thick lines denote keyword's putative endpoints.

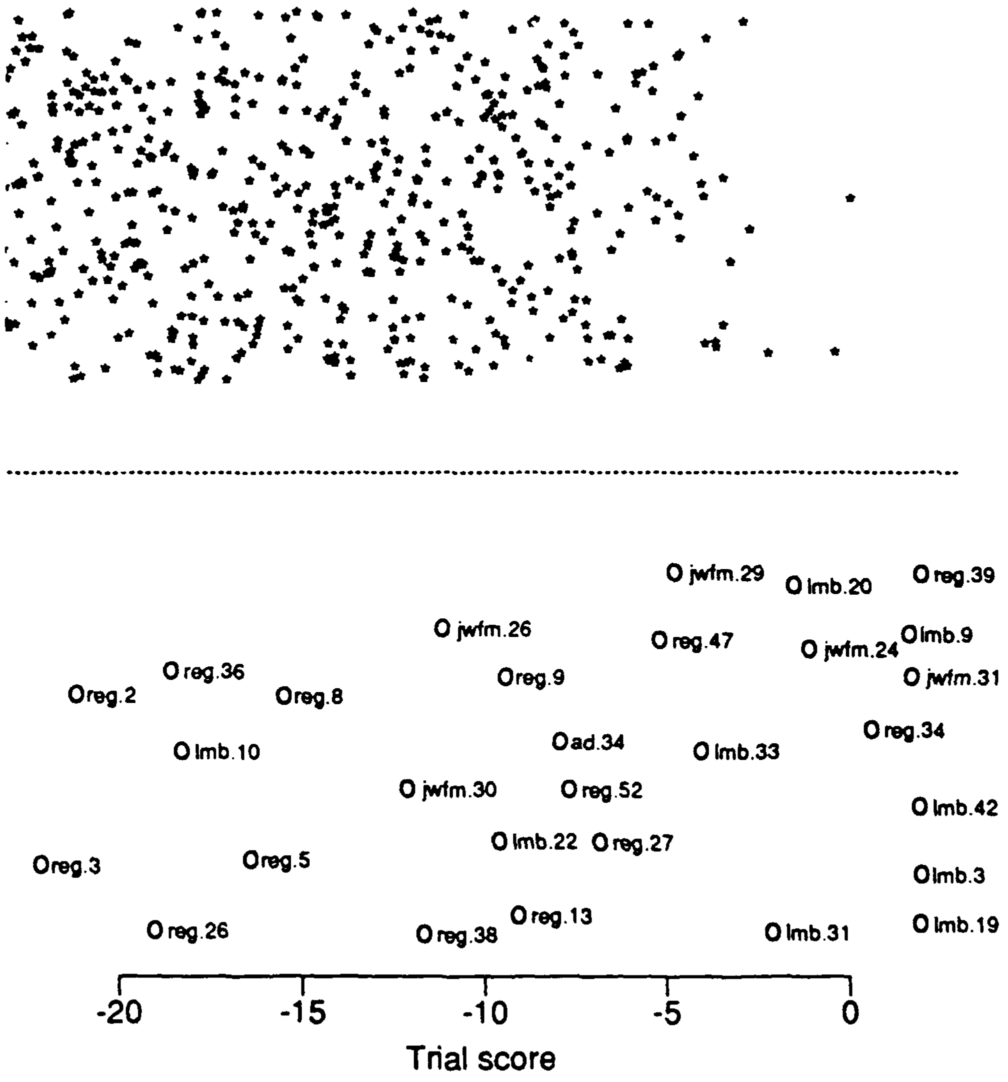


Figure 6.4: Trial score scatter plot for "near." The asterisks (*) represent garbage trials and the o's represent keyword trials. The garbage trial and keyword trial points occupy the top and bottom halves of the y-axis and their positions along this axis are generated randomly. The letters in each utterance label are the speaker's initials and the number is the utterance number for that speaker. Only garbage trial scores that are higher than the lowest keyword trial score are plotted.

exceed the lowest keyword trial score are plotted. The garbage and keyword trial points occupy the top and bottom halves of the y-axis respectively and their positions along this axis are generated randomly.

The user can interact with the plot in one of two ways. In "identify" mode, clicking on a point causes it to be labelled with an utterance identifier. On the illustrated plot, each identifier consists of the speaker initials and utterance number. This is useful for identifying utterances for further study and for discovering patterns. In particular, since the identifier includes a tag for the speaker name, the plot can be used to discover speaker-dependencies.

Once the user has identified points for further study, he/she can invoke "explore" *mode*. In this mode, clicking on a point brings up a segment score plot for the represented trial.

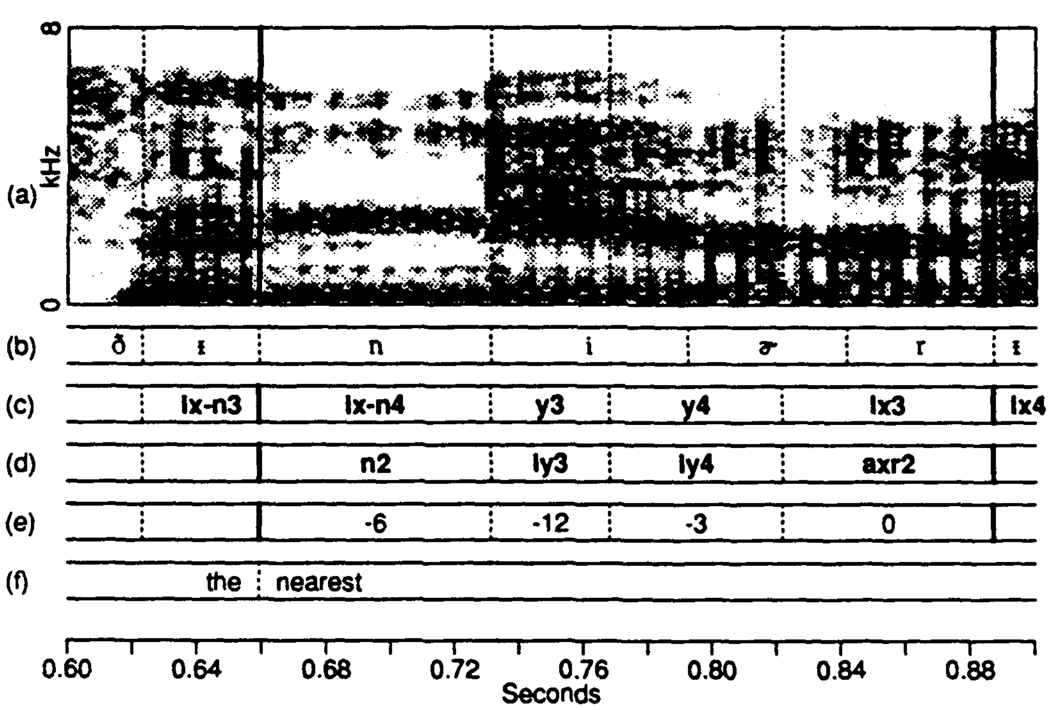
We labelled each keyword trial on the plot with its identifier. It can be seen tnat while most of the lowest keyword trial scores are due to speaker "reg", few of the high ones are. Thus, spotter performance appears to be highly speaker-dependent.

The lowest scoring trial is labelled "reg.3." In "explore" mode, we clicked on this point and brought up the bottom plot of Figure 6.3 so as to investigate the misalignment problem in more detail.

6.2.3 Forced Alignments

One shortcoming of the segment score plot is that it displays only the highest scoring alignment between the keyword model and the utterance. For a situation such as that being discussed here, it is more important to know how segments would be scored if the keyword model was properly aligned to the keyword. This knowledge would enable the designer to determine why the improper alignment scores higher than the proper one. To generate this information, we "force" the spotter to use a path that is aligned with the keyword and display a segment score plot for this alignment, as shown in Figure 6.5.

The plot is implemented by making the segment score plot interactive.



·Figure 6.5: Forced path segment score plot for phone-trained "near" spotter. (a) Spectrogram. (b) Phonetic transcription. (c) State sequence through garbage-keyword network. (d) *Forced* state sequence through keyword model. (e) Relative segment scores for forced path. (f) Orthographic transcription. Times are measured from beginning of utterance.

Clicking at the segment boundaries most closely aligned to the word's endpoints causes the spotter to compute the best path possessing those endpoints. In this case, we specified the end of "near" to be at the segment boundary closest to to the er^d of the /r/ in "nearest." The path is then added to the segment score plot. For clarity, the figure includes the segment score plot for the forced path only.

From the plot, it can be seen that both of the first two segments are poorly matched to the model. We will consider each of these segments in turn. The score for the segment associated with the /n/ in "near" is higher on the garbage **ix-n** model than on the keyword's **n** model. The **ix-n** model is actually matching the correct pair of phones. However, the keyword model is deficient in that it does not include the **ix-n** subword model. In general, the full pronunciation networks of Chapter 5 allowed biphone models to account for segmenter behavior within but not across words. Thus, only the **n** model was allowed at the beginning of the word. The finding indicates the spotter might be improved by allowing biphone models whose right label is **n** at the beginning of "near." Determining whether this is so would require further experimentation since, as we showed in Chapter 5, word models can allow too much, as well as too little, variability in pronunciation.

The second segment's relative score is even worse than the first's. For this segment, the word's **iy** subword model is aligned to the /i^y/ in the utterance. This would be fine except that the segment's score on the **y** model is substantially higher. Thus, the /i^y/ is being recognized as a /y/. The spectrogram does not reveal anything unusual about the /i^y/ . Thus, the segment score plot is useful for identifying the segment a being low-scoring but not for characterizing, at a deeper level, why the confusion occurs.

In particular, to apply acoustic-phonetic knowledge so as to improve the models, it would be useful to characterize confusions in the space of acoustic measurements. In this way, areas where models must be improved can be identified. We outline techniques for doing this in the next few sections.

6.2.4 Decomposing the Relative Score: The Diagonal Covariance Case

If one can decompose a segment's relative score into subscores for individual measurements, one can determine which of the measurements are most responsible for a phonetic confusion and focus one's attention on those. It turns out that this decomposition is straightforward when the state PDF's are modelled as diagonal covariance matrices. Thus, we outline the method for this case first. In this discussion, we are use the term "measurements" loosely to include linear transformations of measurements as well.

Let $\boldsymbol{y}$ be the segment's observation vector, and q its length. Then, according to Equation 4.12, the keyword state score λ_W is computed as

$$\lambda_W = -\frac{q}{2}\log 2\pi - \frac{1}{2}\sum_{j=1}^{q}\{[(y_j - m_{Wj})v_{Wj}]^2 - \log v_{Wj}\} \tag{6.1}$$

as derived in Section 4.8.3. Likewise, the garbage model state that best matches the segment can be represented by a mean vector $\boldsymbol{m}_G$ and a vector of weights $\boldsymbol{v}_G$. The score λ_G of the segment on the garbage model state is then

$$\lambda_G = -\frac{q}{2}\log 2\pi - \frac{1}{2}\sum_{j=1}^{q}\{[(y_j - m_{Gj})v_{Gj}]^2 - \log v_{Gj}\} \tag{6.2}$$

Letting

$$\theta_j = [v_{Gj}(y_j - m_{Gj})]^2 - [v_{Wj}(y_j - m_{Wj})]^2 - \log\frac{v_{Gj}}{v_{Wj}},\ 1 \leq j \leq q, \tag{6.3}$$

the relative score $\Theta = \lambda_W - \lambda_G$ can thus be expressed as

$$\Theta = \sum_{j=1}^{q}\theta_j. \tag{6.4}$$

Thus θ_j is the share of the relative score due to the j^{th} measurement. We will refer to this as the j^{th} relative score share. Measurements with negative shares contribute to a negative relative score and are thus most responsible for a phonetic confusion.

The score decomposition is displayed as a profile plot [Chambers 83, p.163] for which each measurement being profiled is labelled by its name. The user generates the plot by clicking on the appropriate segment in the segment score plot. An example is displayed in Figure 6.6. The figure pertains to the segment in Figure 6.5 that was associated with an /i^y/ but was recognized as a /y/. The measurements are the 39 discriminants, whose PDF is modelled with a diagonal covariance matrix for each state. The measurement name $a.\mathrm{D}b$ refers to the b^{th} GMDA discriminant computed on the group abbreviated a. The group names and abbreviations are defined in Table 6.1. Note that bars to the left of the vertical axis correspond to measurements contributing to the confusion.

Abbreviation	Group
vow	vowels
nas	nasals
stp	stops
frc	fricatives
clo	closures
nas.vow	nasals/vowels (between-group)
stp.clo.frc	stops/closures/fricatives (between-group)
nas.clo	nasals/closures (between-group)

Table 6.1: Abbreviations of discriminants in observation vector.

The plot indicates that the fifth and tenth vowel discriminants (abbreviated as "vow.D.5" and "vow.D.10") contribute most to the confusion, although almost all of the discriminants contribute to some extent. We observed similar score decomposition profiles for the other low-scoring tokens uttered by speaker "reg" as well. This indicated that the confusions might all be related to the same model deficiency.

However, it is unclear from this information how to pursue the investigation. First of all, the scores on many of the discriminants contribute to the error. Thus, it is difficult to identify a small number of measurements for

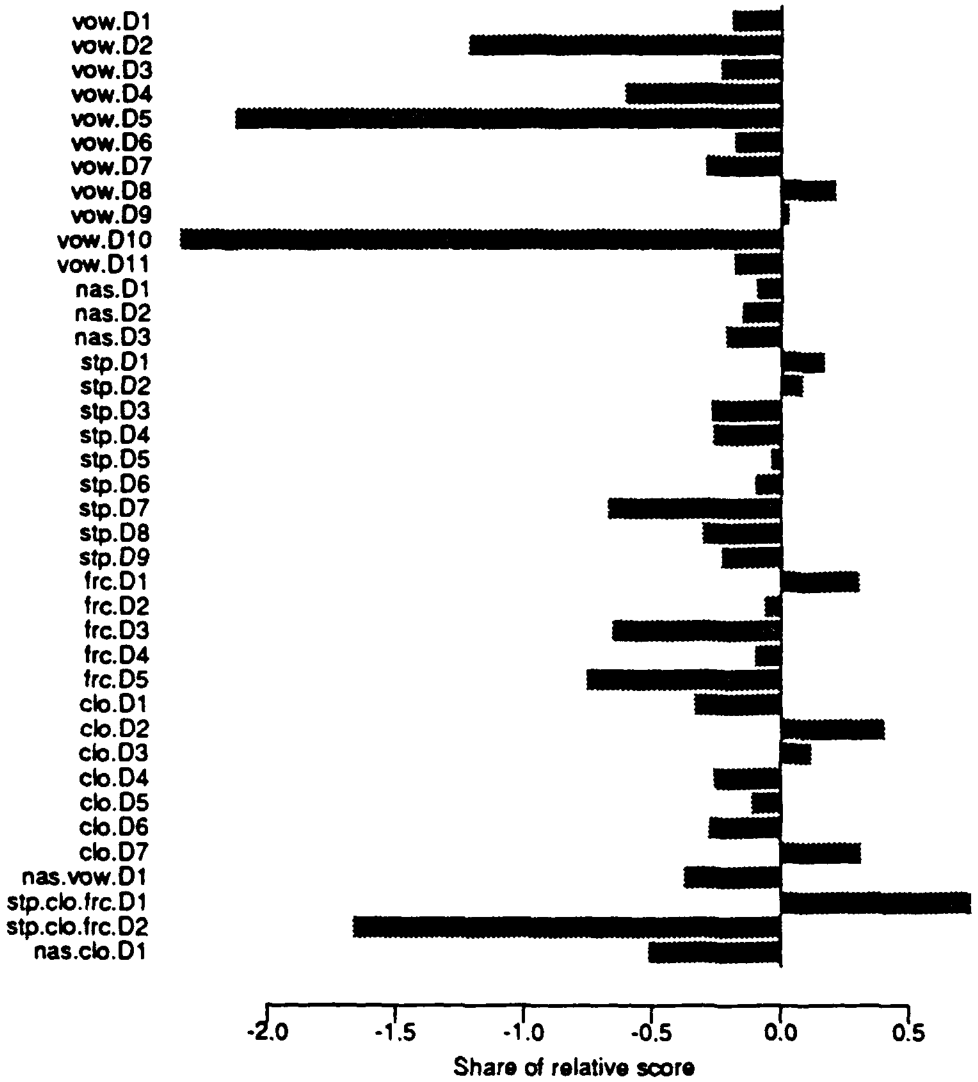


Figure 6.6: Relative score decomposition profile - diagonal covariance case. Barplot labels are names of discriminants. Explanation of names in text. Bars to the left of the vertical axis correspond to discriminants contributing to the confusion. For this case, the segment was associated with an $/i^y/$, and the keyword and garbage subword model states are **iy3** and **y3** respectively.

further study. Additionally, while some of the lower-index discriminants are similar to distinctive feature values (see Section 4.7.3), the higher ones are not easily interpretable.

A better approach might be to decompose the relative score in a domain that is more interpretable. The untransformed measurements, which in most speech recognizers are based on spectral representations such as MFSC's, are good candidates for such an approach, since speech scientists are accustomed to working with spectral representations. In the next section we develop a technique for decomposing the score in the domain of the original measurements.

6.2.5 Decomposing the Relative Score in the Untransformed Measurement Domain

Derivation

The decomposition of the score in the original measurement space is not straightforward because the measurement covariance matrix is not assumed to be diagonal. However, because the segment scores are based on linear transformations of the original measurements, a generalization of the method is possible.

We begin the derivation by rewriting Equation 6.1 in matrix notation:

$$\lambda_W = -\frac{q}{2}\log 2\pi - (\boldsymbol{y} - \boldsymbol{m}_W)^T \boldsymbol{V}_W (\boldsymbol{y} - \boldsymbol{m}_W) + \frac{1}{2}\log\det(\boldsymbol{V}_W) \qquad (6.5)$$

where $\boldsymbol{V}_W = \mathrm{diag}\,(v_{W1}^2, \ldots, v_{Wq}^2)$ is a $q \times q$ diagonal matrix whose elements are squares of the keyword model state's weights. Note that this expression is valid for any multivariate Gaussian PDF model, not just one with a diagonal covariance matrix. In the general case, $\boldsymbol{V}_W = \boldsymbol{\Sigma}_W{}^{-1}$, where $\boldsymbol{\Sigma}_W$ is the state covariance matrix.

It will turn out to be convenient to rewrite Equation 6.5 as

$$\lambda_W = -\frac{q}{2}\log 2\pi - D_W + Z_W \qquad (6.6)$$

where

$$D_W = (\boldsymbol{y} - \boldsymbol{m}_W)^T \boldsymbol{V}_W (\boldsymbol{y} - \boldsymbol{m}_W). \tag{6.7}$$

and

$$Z_W = \frac{1}{2} \log \det(\boldsymbol{V}_W). \tag{6.8}$$

The first step is to decompose D_W in the space of the original measurements. This quantity is a type of distance between the measurement vector and the state mean and larger values of it imply lower scores. Let $\boldsymbol{A}$ be the matrix that transforms the vector of measurements $\boldsymbol{x}$ to the observation vector $\boldsymbol{y}$. In our case, $\boldsymbol{A}$ is the product of two transformation matrices: the principal component transformation of the MFSC's to the MFSPC's and the transformation of the MFSPC's and other measurements to the discriminants. If the size of the measurement set is p, $\boldsymbol{A}$ is a $p \times q$ matrix with $p > q$ since the transformation acts to reduce dimensionality. We will continue with the convention introduced in Section 1.3 of expressing measurement and observation vectors as row vectors. Thus,

$$\boldsymbol{y}^T = \boldsymbol{x}^T \boldsymbol{A} \tag{6.9}$$

and because the mean of a set of vectors is a linear operation,

$$\boldsymbol{m}_W^T = \boldsymbol{\mu}_W^T \boldsymbol{A} \tag{6.10}$$

where $\boldsymbol{\mu}_W$ is the mean of the measurement vectors used to train the keyword state. In practice, we do not use the actual $\boldsymbol{x}$ and $\boldsymbol{\mu}_W$ in the computation, for reasons to be discussed below. Instead we use estimates $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{\mu}}_W$. For the sake of clarity, we will postpone the definition of these estimates until later.

As we show in the definition of the estimates,

$$\boldsymbol{A}^T(\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}_W) = \boldsymbol{y} - \boldsymbol{m}_W. \tag{6.11}$$

Thus, combining Equations 6.7 and 6.11 yields

$$D_W = (\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}_W)^T \boldsymbol{A} \boldsymbol{V}_W \boldsymbol{A}^T (\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}_W) \tag{6.12}$$

which can in turn be expressed as

$$D_W = \boldsymbol{d}^T \boldsymbol{U} \boldsymbol{d} \tag{6.13}$$

where

$$\boldsymbol{d} = \hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}_W \tag{6.14}$$

and

$$\boldsymbol{U} = \boldsymbol{A}\boldsymbol{V}_W\boldsymbol{A}^T \tag{6.15}$$

by definition.

The key idea in the derivation is to assign each measurement an appropriate share of the distance D_W. Toward this end, we express the matrix multiplication of Equation 6.13 in scalar form:

$$D_W = \sum_{k=1}^{p}\sum_{j=1}^{p}(d_k u_{kj} d_j) \tag{6.16}$$

where d_k, d_j and u_{kj} are elements of $\boldsymbol{d}$ and $\boldsymbol{U}$.

Each term in Equation 6.16 for which $k \neq j$ is the product of two measurements' distances from their state means d_k and d_j and the multiplier u_{kj}. Thus the size of each such element cannot be uniquely attributed to a single measurement's distance from its mean. This is what makes the problem more difficult than the diagonal covariance case, for which the score could be decomposed into terms that could each be attributed to a single measurement.

One alternative would be to set the j^{th} share of the distance to be the sum of all terms involving d_j. However, in this case, the terms $d_k u_{kj} d_j$ and $d_j u_{jk} d_k$ are included in both the k^{th} and j^{th} shares so that when the shares are summed both terms are counted twice. Thus, the sum of the shares is not D_W. This is undesirable since the whole point of the decomposition is to evaluate each measurement's contribution to the total segment score.

A better alternative makes use of the fact that $u_{kj} = u_{jk}$, i.e., $\boldsymbol{U}$ is symmetric. This is true because $\boldsymbol{V}_W$ is diagonal and therefore symmetric, and thus $\boldsymbol{U}$ is symmetric, too. Thus, $d_k u_{kj} d_j = d_j u_{jk} d_k$. Thus, it seems reasonable to include one of these terms in the j^{th} distance share and the other in the k^{th}. In this way each term is counted only once when the shares are summed and so the shares add to D_W. Also, $d_k u_{kj} d_j + d_j u_{jk} d_k$, the joint contribution of the two measurements to the distance, is split equally across both shares, which seems reasonable. Note that the argument is valid as long as $\boldsymbol{V}_W$ is symmetric and thus holds in general since $\boldsymbol{V}_W$ is a covariance matrix.

The distance share e_{W_k} for measurement k is thus defined as

$$e_{W_k} = d_k \sum_{j=1}^{p} u_{kj} d_j, \ 1 \leq k \leq p. \tag{6.17}$$

If Equation 6.16 is re-expressed as

$$D_W = \sum_{k=1}^{p} d_k \sum_{j=1}^{p} (u_{kj} d_j) \tag{6.18}$$

it is clear that

$$D_W = \sum_{k=1}^{p} e_{W_k}, \tag{6.19}$$

as claimed. Note that $d_k = 0$ implies $e_{W_k} = 0$, so that when a measurement is equal to its state mean, its share of the distance is 0. Also note that if $\boldsymbol{U}$ is diagonal, implying no correlation among measurements, $e_{W_k} = (u_{kk} d_k)^2$, so the share is proportional to the squared difference between the measurement and its mean. These are intuitively appealing qualities for the distance share to have.

Equation 6.17 can be written compactly in matrix notation:

$$\boldsymbol{e}_W = \operatorname{diag}\left[(\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}_W)^T\right] \boldsymbol{A} \boldsymbol{V}_W \boldsymbol{A}^T (\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}_W) \tag{6.20}$$

where we have substituted the definitions of $\boldsymbol{d}$ from Equation 6.14 and $\boldsymbol{U}$ from Equation 6.15 in the expression. An analogous expression can be derived for $\boldsymbol{e}_G$, the vector of distance shares for the garbage state. Letting $\boldsymbol{V}_G = \operatorname{diag}(v_{G1}^2, \ldots, v_{Gq}^2)$,

$$\boldsymbol{e}_G = \operatorname{diag}\left[(\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}_G)^T\right] \boldsymbol{A} \boldsymbol{V}_G \boldsymbol{A}^T (\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}_G) \tag{6.21}$$

It is worthwhile to consider the case where D_W is a Mahalanobis distance and $p = 2$ so that

$$D_W = \boldsymbol{d}\boldsymbol{C}^{-1}\boldsymbol{d}$$

where $\boldsymbol{C}$ is the estimated covariance matrix for measurements x_1 and x_2. Letting σ_k be the estimated standard deviation for $x_k, i = 1,2$ and ρ be the correlation coefficient between the measurements, it is easy to show that

$$e_{W1} = \frac{1}{2(1-\rho^2)}[\frac{d_1}{\sigma_1}^2 - \rho\frac{d_1}{\sigma_1}\frac{d_2}{\sigma_2}]$$

and

$$e_{W2} = \frac{1}{2(1-\rho^2)}[\frac{d_2}{\sigma_2}^2 - \rho\frac{d_1}{\sigma_1}\frac{d_2}{\sigma_2}].$$

Thus, the first term in each distance share is due totally to the deviation of the measurement from its state mean and and the second term reflects the contribution to the distance from the relationship of d_1 to d_2. The total distance accounted for by these terms is proportional to $2\rho(d_1/\sigma_1)(d_2/\sigma_2)$ and it is divided equally between both measurement's shares, as discussed above.

To continue the decomposition of λ_W, Z_W, the term related to the determinant of $\boldsymbol{V}_W$ must be decomposed in the measurement space. The decomposition is intuitive rather then formal. First, compare Equation 6.1 to Equation 6.5. They are the same equations but the former describes the decomposition of λ_W in the diagonal covariance case and the latter in the general case. The equations show that the share of Z_W assigned to measurement j in the diagonal case is $\log v_{Wj}$. Thus, this share of Z_W is proportional to the logarithm of the weight put on measurement j. Analogously, the share of Z_W assigned to measurement k in the measurement space should be proportional to the logarithm of its weight in the computation of λ_W. Comparing Equation 6.12 to Equation 6.5, it can be seen that $\boldsymbol{A}\boldsymbol{V}_W\boldsymbol{A}^T$ plays the role of the weighting matrix in the measurement space. As above, we will abbreviate this matrix as $\boldsymbol{U}$. Because $\boldsymbol{U}$ is not diagonal, it is not clear how to derive measurement-specific weights from it. However, it can be seen from Equation 6.16 that each diagonal element u_{kk} acts to weight the term ${d_k}^2$ and so the best choice seems to be the diagonal elements of $\boldsymbol{U}$. Thus, z_{Wk}, the share of Z_W assigned to each measurement k is set to be proportional to $\log u_{kk}$. Thus

$$z_{Wk} = \frac{1}{2}\log\det(\boldsymbol{V}_W)\frac{\log u_{kk}}{\sum_{k=1}^{p}\log u_{kk}},\ 1 \leq k \leq p. \tag{6.22}$$

In vector form, this can be written

$$\boldsymbol{z}_W = \frac{1}{2}\log\det(\boldsymbol{V}_W)\frac{[\log[\boldsymbol{A}\boldsymbol{V}_W\boldsymbol{A}^T]_{11},\ldots,\log(\boldsymbol{A}\boldsymbol{V}_W\boldsymbol{A}^T)_{pp}]^T}{\sum_{k=1}^{p}\log[\boldsymbol{A}\boldsymbol{V}_W\boldsymbol{A}^T]_{kk}}. \tag{6.23}$$

Analogously,

$$\boldsymbol{z}_G = \frac{1}{2}\log\det(\boldsymbol{V}_G)\frac{[\log[\boldsymbol{A}\boldsymbol{V}_G\boldsymbol{A}^T]_{11},\ldots,\log(\boldsymbol{A}\boldsymbol{V}_G\boldsymbol{A}^T)_{pp}]^T}{\sum_{k=1}^{p}\log[\boldsymbol{A}\boldsymbol{V}_G\boldsymbol{A}^T]_{kk}}. \tag{6.24}$$

Equations 6.17 and 6.22 define the decompositions of the last two terms in Equation 6.5 in the space of the measurements. Combining them with Equation 6.5 yields

$$\lambda_W = -\frac{q}{2}\log 2\pi + \sum_{k=1}^{p}(z_{Wk} - e_{Wk}). \tag{6.25}$$

Analogously,

$$\lambda_G = -\frac{q}{2}\log 2\pi + \sum_{k=1}^{p}(z_{Gk} - e_{Gk}). \tag{6.26}$$

Thus, letting

$$\phi_k = [(z_{Wk} - e_{Wk}) - (z_{Gk} - e_{Gk})],\ 1 \leq k \leq p, \tag{6.27}$$

$$\Theta = \lambda_W - \lambda_G = \sum_{k=1}^{p}\phi_k \tag{6.28}$$

and ϕ_k is measurement k's share of the relative score Θ.

To complete the derivation, we justify our use of the estimates $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{\mu}}_W$ and define them. In practice, we could use the values $\boldsymbol{x}$ and $\boldsymbol{\mu}_W$ in the relative score decomposition. The first is simply the test segment's measurement vector. The second is the mean of the measurement vectors that were used to train the keyword state. This could be stored for the purposes of the computation. However, the word spotter itself cannot reconstruct the measurement vector once its dimensionality has been reduced by the transformation $\boldsymbol{A}$, since this would necessitate finding a matrix $\boldsymbol{A}_R^{\mathrm{inv}}$ such that

$$\boldsymbol{x}^T = \boldsymbol{y}^T \boldsymbol{A}_R^{\mathrm{inv}} = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{A}_R^{\mathrm{inv}}. \tag{6.29}$$

The notation signifies the fact that $\boldsymbol{A}_R^{\mathrm{inv}}$ is a right inverse of $\boldsymbol{A}$. However, it can be shown that solving for $\boldsymbol{A}_R^{\mathrm{inv}}$ is the equivalent of solving p equations in q unknowns and since $p > q$, there is no general solution. This is true in general for dimensionality reduction techniques. By the same reasoning, the spotter makes no use of the $\boldsymbol{\mu}_W$ in computing segment scores since this vector cannot be reconstructed from $\boldsymbol{m}_W$.

Thus, an analysis of the relative score based on the untransformed segment and mean measurement vectors confounds two processes: dimensionality reduction and acoustic modelling in the transformed measurement space. The analysis we propose deals with only the second of these processes and so is conceptually cleaner, in our view.

While the untransformed measurement and mean vectors $\boldsymbol{x}$ and $\boldsymbol{\mu}_W$ are unavailable to the spotter, we assume that they are represented in the spotter by their least-squares estimates given the transformed vectors $\boldsymbol{y}$ and $\boldsymbol{m}_W$. Thus, we use these estimates when decomposing the segment relative score.

We will derive the estimate $\hat{\boldsymbol{x}}$ of the measurement vector. The estimate of the state mean is derived in a similar manner. Consider the least-squares estimate $\hat{x}_j$ of a single element of $\boldsymbol{x}$ given $\boldsymbol{y}$. The goal is to find a linear transformation

$$\hat{x}_j = c_j + \boldsymbol{y}^T b_j$$

that minimizes the expected value of $x_j - \hat{x}_j{}^2$. This is of the form of a multiple regression problem for which $\boldsymbol{y}$ is the vector of regressors and x_j is the response variable. The regression coefficients must be determined for all j, $1 \leq j \leq p$. Thus, the problem can be framed as a multivariate multiple regression problem [Johnson 88]

$$\hat{\boldsymbol{x}}^T = \boldsymbol{c}^T + \boldsymbol{y}^T \boldsymbol{B} \tag{6.30}$$

where $\boldsymbol{B} = [\boldsymbol{b}_1 \ldots \boldsymbol{b}_p]$ and $\boldsymbol{c}^T = (c_1, \ldots, c_p)^T$.

To determine the coefficients B, a training set of regressors and explanatory variables is needed. Let

$$\boldsymbol{Y} = \begin{bmatrix} y_{11} & \cdots & y_{iq} \\ \vdots & & \vdots \\ y_{n1} & \cdots & y_{nq} \end{bmatrix}$$

be the training set of regressors, centered so that

$$\sum_{i=1}^{n} y_{ij} = 0, \ 1 \leq j \leq p$$

and

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{ip} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}$$

be the corresponding vectors of explanatory variables, also centered. Then, the least-squares solution is

$$\boldsymbol{B} = (\boldsymbol{Y}^T\boldsymbol{Y})^{-1}\boldsymbol{Y}^T\boldsymbol{X}. \tag{6.31}$$

and

$$\mathbf{c} = \boldsymbol{m}_x - \boldsymbol{B}^T\boldsymbol{m}_y \tag{6.32}$$

where $\boldsymbol{m}_x = [1, \ldots, 1]\boldsymbol{X}$ and $\boldsymbol{m}_y = [1, \ldots, 1]\boldsymbol{Y}$, i.e., they are the sample mean explanatory variable and regressor vectors.

However, $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent untransformed and transformed measurement data, respectively, and

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{A}. \tag{6.33}$$

This is true even when the data are centered. Thus, combining Equations 6.31 and 6.33,[1]

$$\boldsymbol{B} = (\boldsymbol{A}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{X}^T\boldsymbol{X} \tag{6.34}$$

and combining Equations 6.32 and 6.33,

$$\boldsymbol{c} = \boldsymbol{m}_x[\boldsymbol{I} - (\boldsymbol{A}\boldsymbol{B})^T]. \tag{6.35}$$

[1] It can be shown that the back-transformation of the principal component transformation that was used to generate Figure 2.1 is a special case of these equations.

These equations are used to estimate the segment's measurement vector and the state mean in the space of untransformed measurements. Thus,

$$\hat{\boldsymbol{x}} = \boldsymbol{B}^T \boldsymbol{y} + \boldsymbol{c} \tag{6.36}$$

and

$$\hat{\boldsymbol{\mu}}_W = \boldsymbol{B}^T \boldsymbol{m}_W + \boldsymbol{c}. \tag{6.37}$$

The identical method is used to estimate the state mean on the garbage state. As in Section 2.3.3, we refer to $\boldsymbol{B}$ as the back-transform matrix since it attempts to reverse the transformation of the measurements.

To verify the claim of Equation 6.11, we combine Equations 6.34,6.36 and 6.37 to get

$$\begin{aligned} \boldsymbol{A}^T(\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}_W) &= \boldsymbol{A}^T \boldsymbol{B}^T (\boldsymbol{y} - \boldsymbol{m}_W) \qquad (6.38) \\ &= \boldsymbol{y} - \boldsymbol{m}_W, \end{aligned}$$

as claimed.

It is necessary to specify untransformed measurement data $\boldsymbol{X}$ to determine $\boldsymbol{B}$ and $\boldsymbol{c}$. The most reasonable data set for this purpose is the one that was used to determine the transformation $\boldsymbol{A}$ in the first place. Thus, in our case, the data consisted of the composite segments defined in Chapter 4 that were used as training data for the multiple discriminant analysis.

Equations 6.20, 6.36 and 6.37 can be combined to yield

$$\boldsymbol{e}_W = \text{diag}\,[(\boldsymbol{y} - \boldsymbol{m}_W)^T \boldsymbol{B}] \boldsymbol{A} \boldsymbol{V}_W (\boldsymbol{y} - \boldsymbol{m}_W) \tag{6.39}$$

where we have used the fact that $\boldsymbol{B}\boldsymbol{A} = I$ to simplify the expression. Analogously,

$$\boldsymbol{e}_G = \text{diag}\,[(\boldsymbol{y} - \boldsymbol{m}_G)^T \boldsymbol{B}] \boldsymbol{A} \boldsymbol{V}_G (\boldsymbol{y} - \boldsymbol{m}_G) \tag{6.40}$$

Elements of the vectors defined by Equations 6.39, 6.40, 6.23 and 6.24 can be substituted into Equation 6.27 to compute the spectral domain error components.

Case Study Findings

We used this method to analyze the "reg.3" trial. The set of measurements for which score shares were computed included

1. predicted formants PF_1, PF_2, and energy (ENGY) at all eight spectral positions (24 measurements),
2. $PPMFSPC_1$, $PMFSPC_1$, $FMFSPC_1$, $FFMFSPC_1$ (4 measurements),
3. differences $DPMFSPC_i = PMFSPC_i - PPMFSPC_i$ and $DFMFSPC_i = FFMFSPC_i - FMFSPC_i$, $2 \leq i \leq 15$ (28 measurements),
4. 40 MFSC's at each of the within-segment positions B, M, A, and E. (160 measurements), and
5. duration

for a total of 217 measurements.

See Section 4.6 for definitions of the abbreviations. It would have been desirable to include only spectral measurements (MFSC's) instead of a mixture of MFSC's and MFSPC's since spectral representations are better matched to our knowledge about speech than are their principal components. However, we were unable to directly use MFSC's of the out-of-segment spectral positions (or even their differences) because of a peculiarity in the way the 117-measurement set used to compute the discriminants was computed. The mathematical details behind this are tedious and will be omitted from the discussion.

Figure 6.7 displays the decomposition in the form of a profile plot. Each of the bars representing predicted formants, energy, MFSPC's and DMFSPC's on the barchart is labelled with its measurement's abbreviation. These appear on the left of the three barcharts. The MFSC's, however, are labelled by their center frequencies, denoted in kHz; to avoid clutter, not all are labelled. The

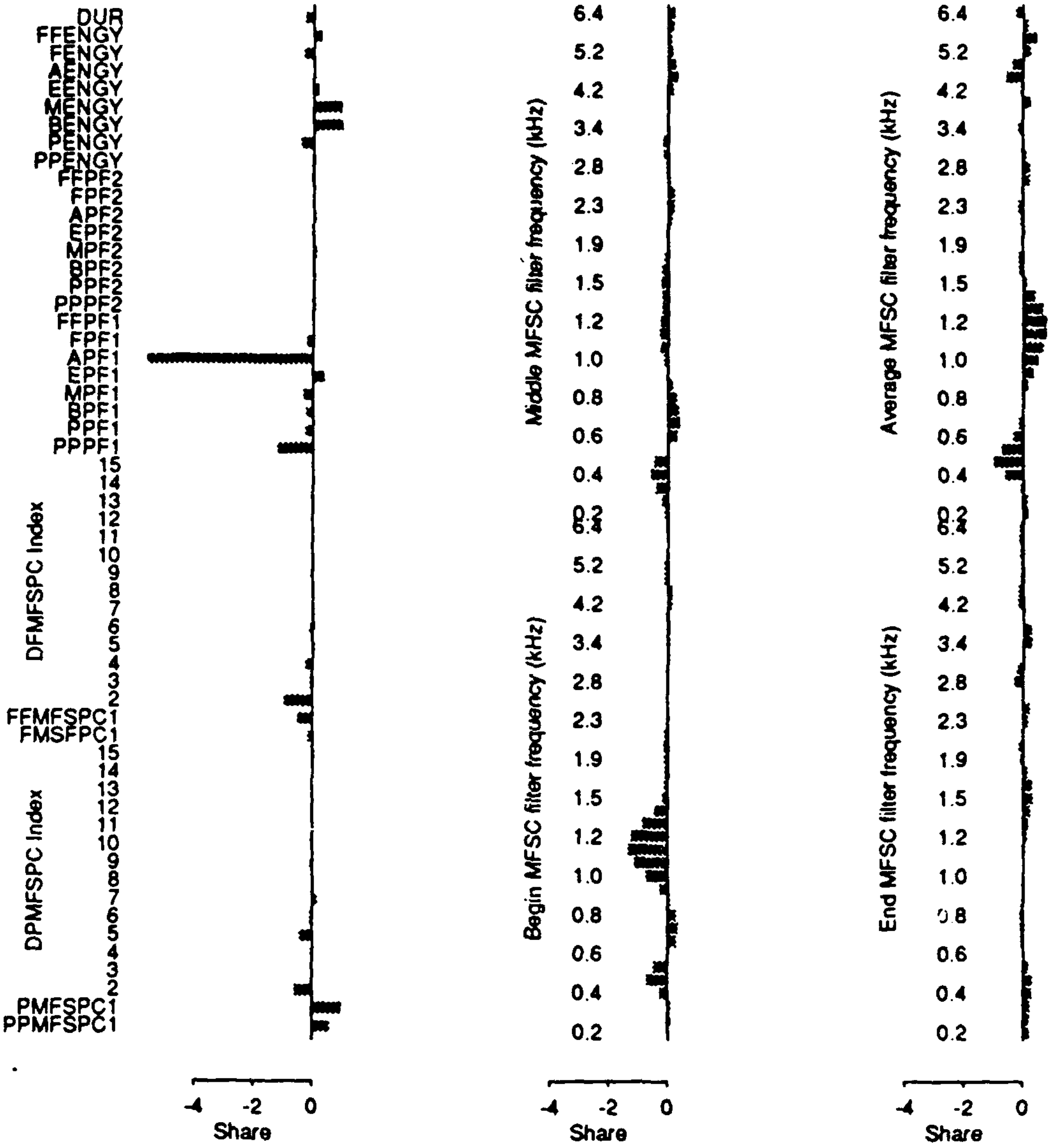


Figure 6.7: Relative score decomposition in untransformed measurement space - "reg.3" example. Left barchart includes components for MFSPC's, predicted formants and energy; each bar is labelled with the measurement's abbreviation, except for DPMSFPC's and DFMFSPC's, which are labelled with their index only. Middle and right barcharts include components for MFSC's, labelled by filter center frequency in kHz; to avoid clutter, not all are labelled. For this segment, the uttered phone is $/i^y/$, and the keyword and garbage model states are **iy3** and **y3**, respectively. The relative score for this segment is -12.5.

position where each set of MFSC's was measured is written vertically to the left of the frequency labels.

The most salient feature of the chart is the high negative value associated with APF_1[2] Almost all the predicted F_1's also have negative error components. This indicates that, although the segment is associated with an /i^{y}/ , its first formant is more closely matched to the **y** model than to the **iy** model. Also, the shares for the MFSC's in the 400-600 Hz range tend to be negative at the beginning, middle and average positions, providing further evidence that the segment's F_1 value is largely responsible for the confusion. The beginning MFSC's in the 1000-1400 Hz range also seem to match the **y** model better although the average MFSC's in this frequency range match the **iy** model better.

To check the consistency of these results, we generated the same plot for several other low-scoring segments associated with /i^{y}/ that were uttered by speaker "reg." A typical example is displayed in Figure 6.8 for the trial from utterance "reg.2." The profile is strikingly similar to that of the previous example. In fact, the six lowest-scoring keyword tokens uttered by this speaker exhibited similar profiles, particularly with respect to the share of the relative score of measurement APF_1.

6.2.6 The Segment/State Profile

Description

One limitation of the score decomposition is that it does not indicate the manner in which each measurement deviates from the correct model. For instance, one cannot tell from Figures 6.7 and 6.8 whether APF_1 is low or high compared to its mean value in the **iy** model. To display this information, we developed the *segment-state profile*, which displays estimates of the segment measurement and keyword and garbage state mean vectors. It also

[2] We will refer to the measurements as if they were the actual ones. However, it should be remembered that they are actually least-square estimates.

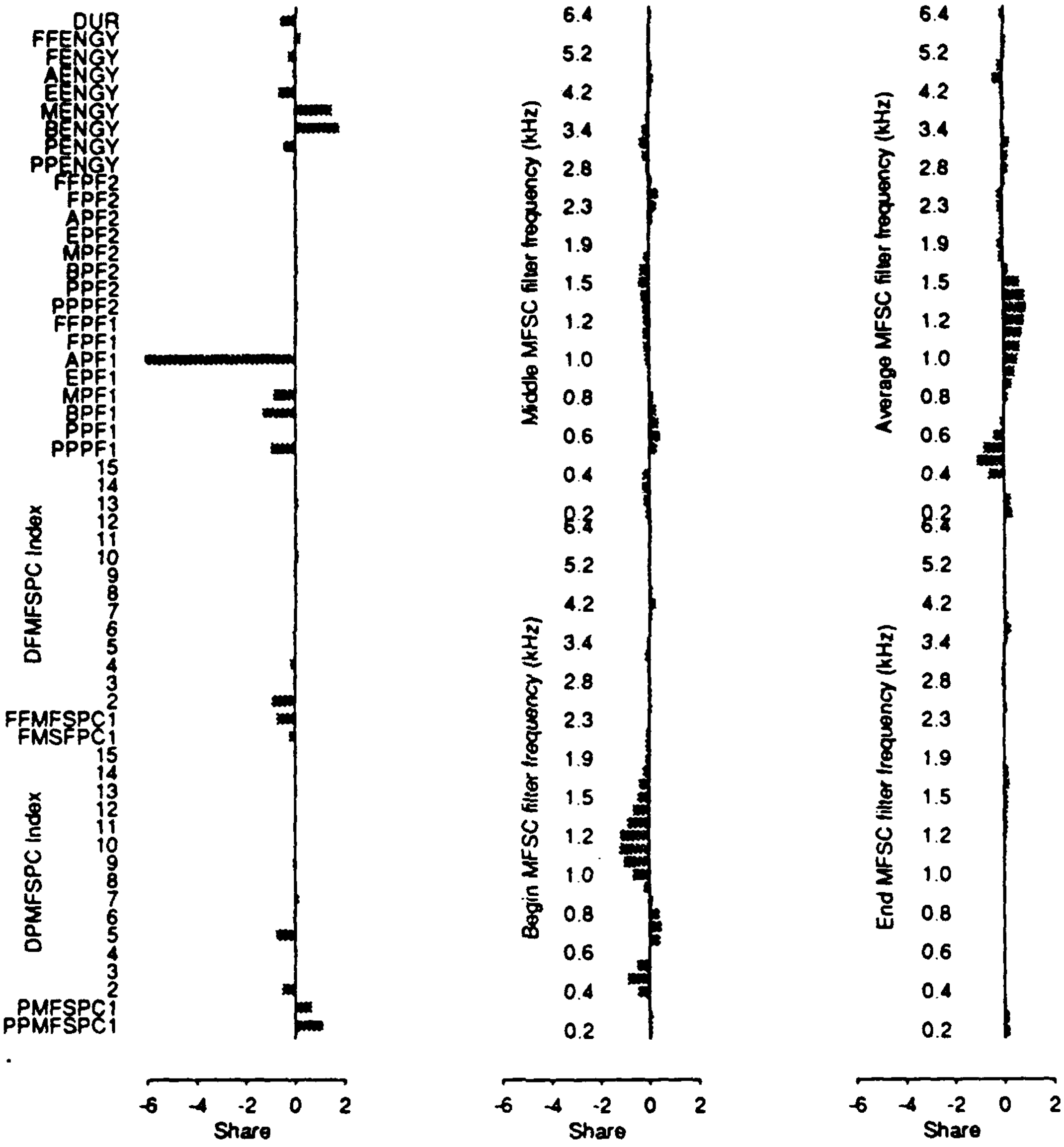


Figure 6.8: Relative score decomposition in untransformed measurement space - "reg.2" example. Left barchart includes components for MFSPC's, predicted formants and energy; each bar is labelled with the measurement's abbreviation, except for DPMSFPC's and DFMFSPC's, which are labelled with their index only. Middle and right barcharts include components for MFSC's, labelled by filter center frequency in kHz; to avoid clutter, not all are labelled. For this segment, the uttered phone is /i^y/ , and the keyword and garbage model states are **iy3** and **y3**, respectively. The relative score for this segment is -15.2.

includes "confidence bounds" around each state mean that provide a scale for ascertaining how much each segment deviates from the mean.

Figure 6.9 displays the segment/state profile for the "reg.3" example. In general, the display consists of a number of panels, each panel pertaining to a subset of the measurements being profiled. Each panel consists of seven profile plots. The profiles are of

1. the least-squares estimate of the segment's measurement vector (solid line in figure),

2. least-squares estimates of the mean vectors for the state that should have best matched the segment and the one that actually did (in this case, the keyword and garbage states, respectively, represented by dotted lines), and

3. estimates of lower and upper "confidence bounds" for the two states in question (represented by "o" and "x" for the keyword and garbage model, respectively).

The estimate of the segment's measurement vector is given by the expression $\boldsymbol{B}_p^T \boldsymbol{y} + \boldsymbol{c}_p$ where $\boldsymbol{y}$, as in the previous section, is the set of transformed measurements, $\boldsymbol{B}_p$ is the matrix for back-transforming them to the profiled measurements, and $\boldsymbol{c}_p$ is the constant term required for the estimate. The values of $\boldsymbol{B}_p$ and $\boldsymbol{c}_p$ are determined similarly to those of $\boldsymbol{B}$ and $\boldsymbol{c}$, as outlined in the previous section. The subscript is used to indicate that the profiled measurements may be different from those of the score decomposition. Thus, they may be associated with a different back-transform matrix and constant term. We discuss this further below. The estimates of the keyword and garbage state means are computed similarly and are given by the expressions $\boldsymbol{B}_p^T \boldsymbol{m}_W + \boldsymbol{c}_p$ and $\boldsymbol{B}_p^T \boldsymbol{m}_G + \boldsymbol{c}_p$.

Finally, the lower and upper confidence bounds for measurement k on the

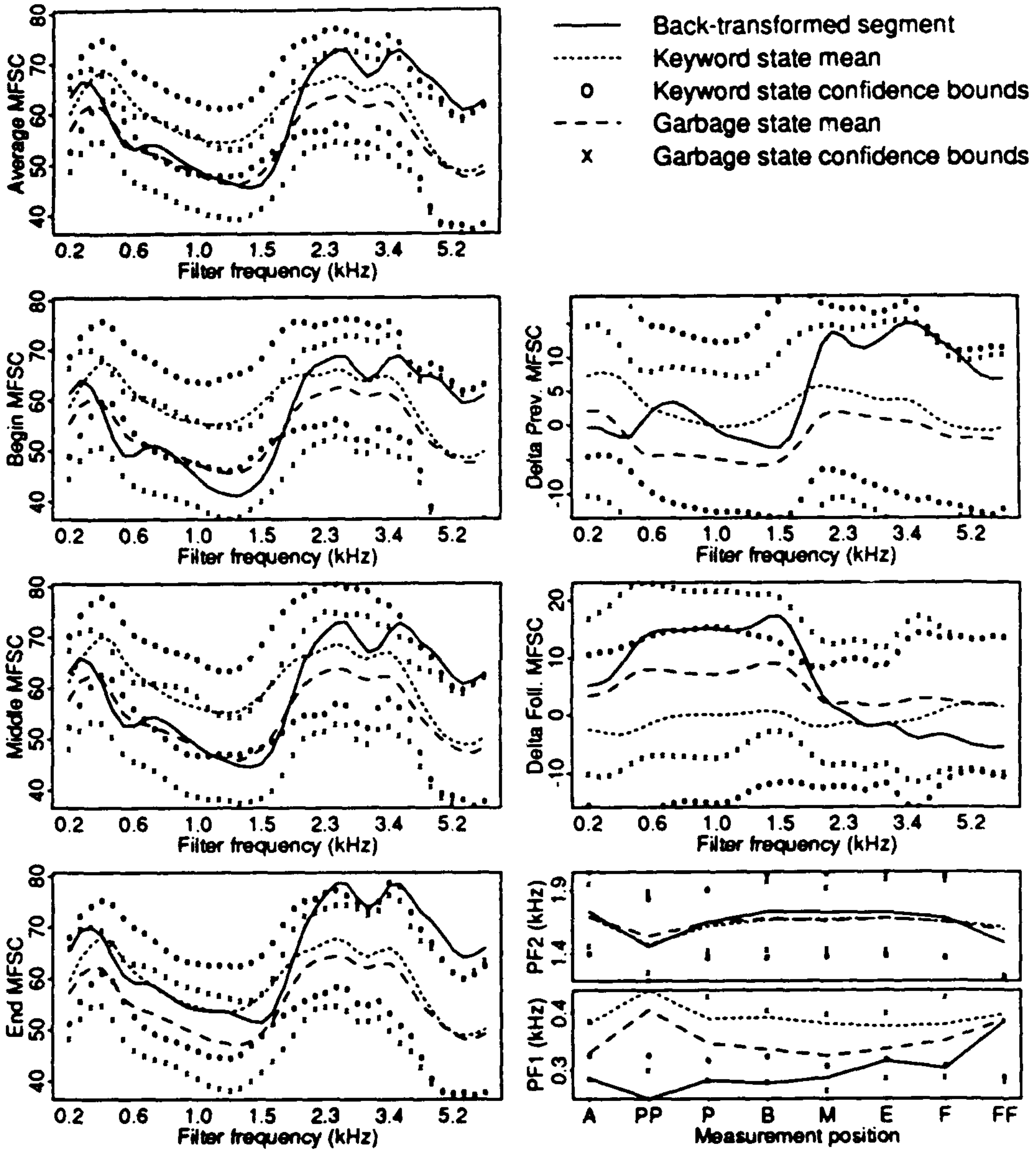


Figure 6.9: Segment/state profile for "reg.3" trial. Each panel displays, according to the legend at top right, the back-transformed segment measurements, state means, and confidence bounds for the measurement set specified by the y-axis label. The top two plots in the right column show changes in the MFSC's between positions PP and P and between positions F and FF respectively. Duration (DUR) and energy (ENGY) panels are omitted to save space. The position along the x-axis of each point in the MFSC plots denotes the center frequency of the corresponding filter. In some panels, several confidence bound points are beyond the plot limits and are not shown.

keyword state are given by

$$(\boldsymbol{B}_p^T \boldsymbol{m}_W + \boldsymbol{c}_p)_k \pm \sqrt{[\boldsymbol{B}_p^T V_W{}^{-1} \boldsymbol{B}_p]_{kk}} \tag{6.41}$$

where V_W, as defined in the previous section, is the inverse of the keyword state PDF's covariance matrix.

The first term in the expression is the estimated state mean and the second its estimated standard deviation for the measurement. To derive the latter, note that $\hat{\boldsymbol{x}}_p = \boldsymbol{B}_p^T \boldsymbol{y}$, where $\hat{\boldsymbol{x}}_p$ is the vector of profiled measurements. Thus,

$$\mathrm{cov}(\hat{\boldsymbol{x}}_p|W) = \boldsymbol{B}_p^T V_W{}^{-1} \boldsymbol{B}_p \tag{6.42}$$

where $\mathrm{cov}(\hat{\boldsymbol{x}}_p|W)$ is the estimated covariance of the measurement vector associated with the keyword state and 6.41 follows. By similar reasoning the confidence bounds for the garbage state are

$$(\boldsymbol{B}_p^T \boldsymbol{m}_G + \boldsymbol{c}_p)_k \pm \sqrt{[\boldsymbol{B}_p^T V_G{}^{-1} \boldsymbol{B}_p]_{kk}} \tag{6.43}$$

The standard deviation is a reasonable scale for ascertaining, for the two states in question, the extent to which each segment measurement deviates from its estimated state mean. Thus, when a segment measurement is outside the band defined by the confidence bounds, it is far from its expected mean on that state and probably contributes to a low score on that state.

Case Study Findings

Before interpreting Figure 6.9 in detail, we point out that the measurements included in the segment-state profile are slightly different from those of the relative score decomposition. In particular, the latter include spectral (MFSC) rather then principal component (MFSPC) differences between positions PP and P and between F and FF (see the top two panels on the right). As we discussed in Section 6.2.5, this is desirable because spectral measurements are easier to interpret. The difference is due to the fact that the peculiarity that prohibited the measurements' inclusion in the score decomposition does not apply to the segment-state profile.

To facilitate the interpretation, we present expanded views of the beginning MFSC and predicted first formant (PF_1) panels in Figure 6.10. We chose these in particular because the relative score decomposition suggested that the PF_1 measurements and MFSC's whose center frequencies were in the 400-600 Hz range were largely responsible for the low relative score on the segment.

It can be seen from Figure 6.10a that the first peak in the segment's spectrum occurs at roughly 250 Hz. This is likely the segment's F_1 at the beginning of the segment. At the same point, the garbage model's mean F_1 is apparently about 350 Hz and the keyword model's is over 400 Hz. Thus, as suggested by the relative score decomposition, the segment's F_1 is closer to the garbage state **y3** mean than to :he keyword model state **iy3** mean. Moreover, while the state mean MFSC's in the range of 400-600 Hz are relatively high due to the effect of a nearby F_1, the segments' are low in this range. The reason for this is that the segment F_1 is apparently well below 400 Hz. In fact, the segment MFSC's in this range are below the word state's confidence band while they are well within the garbage state's. This agrees with the relative score shares, which are slightly negative throughout this range for the beginning MFSC's. Similar behavior can be observed between 1000 and 1800 Hz, again in agreement with the relative score decomposition.

Figure 6.10b shows clearly that the segment's predicted first formant's trajectory remains well below the keyword state's lower confidence bound throughout the segment while remaining at the lower end or just beneath the garbage model's lower bound. Thus, the findings of the relative score decomposition and segment/state profiles strongly suggest that the phonetic confusion between the /i^{y}/ and y is largely due to an unusually low value of F_1 for the segment in question.

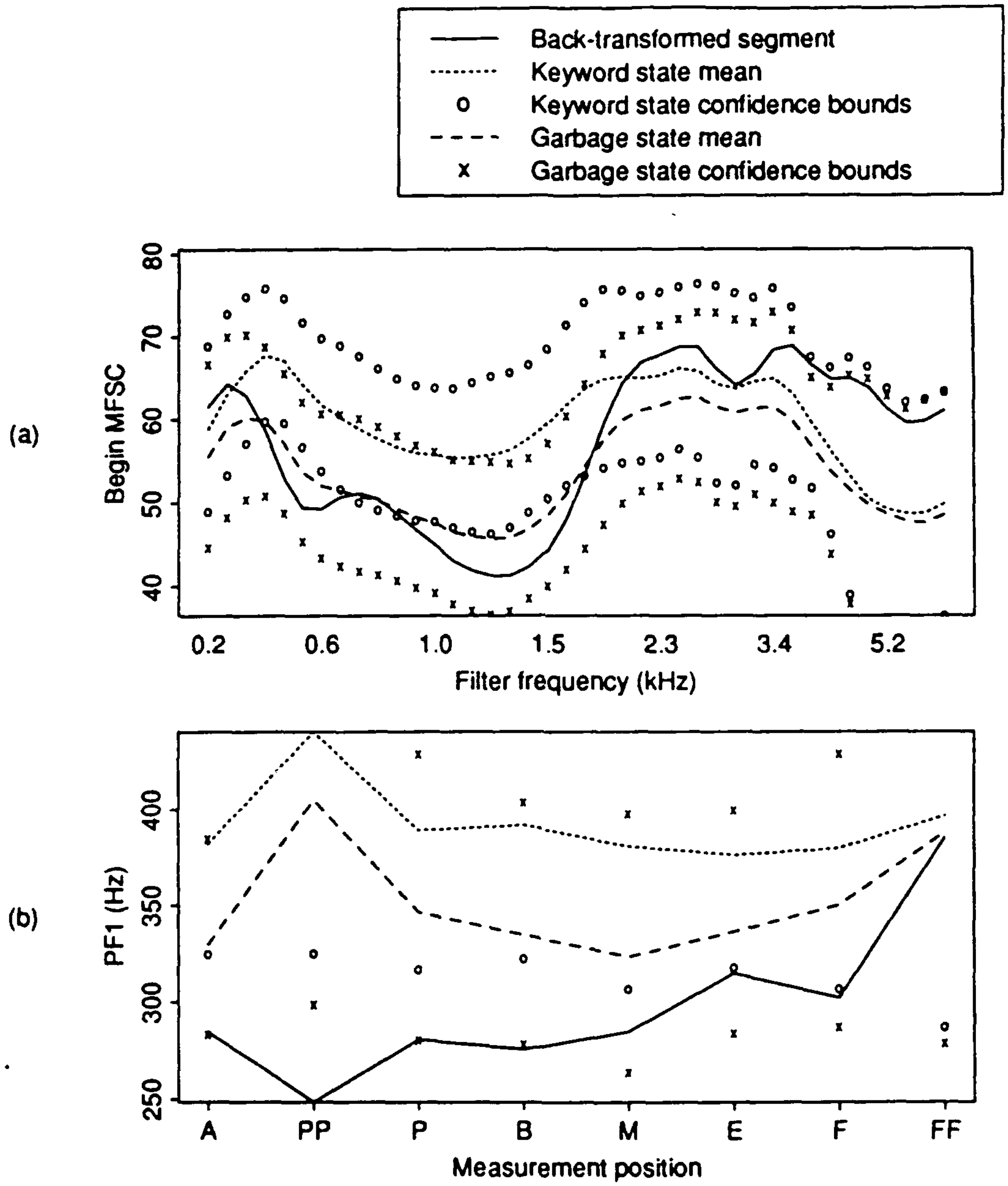


Figure 6.10: Expanded panels of segment/state profiles for (a) beginning MFSC's and (b) predicted first formants (PF_1). Each panel displays, according to the legend, the back-transformed segment measurements, state means, and confidence bounds for the measurement set specified by the y-axis label. The position along the x-axis of each point in the MFSC plots denotes the center frequency of the corresponding filter.

6.2.7 Details of the User Interface for the Relative Score Decomposition and Segment Score Profiles

While we have displayed the relative score decomposition and segment state profiles as separate figures, in our implementation they share a sir gle display on the workstation. The display is summoned by clicking on the segment score plot at the segment to be investigated. The decomposition plot is used to identify measurements responsible for a low score and the segment/state profile is used to look at these measurements more closely. Because the complete segment/state profile, as shown in Figure 6.9, is quite crowded, clicking on a panel summons an expanded view of it, similar to those in Figure 6.10. Also, the segment score profile uses color to disambiguate the individual profile plots that are included in each panel.

It should be emphasized that the segment/state profile takes advantage of the fact that spectral measurements are naturally ordered by frequency. Thus, the individual profile plots in each panel are reasonably easy to interpret, especially if one is accustomed to plots of spectral magnitude. Similarly, the formant estimates are naturally ordered by time and the profiles resemble formant trajectories which are familiar to anyone accustomed to spectrograms or formant trackers. If there were no such ordering in either case, then the problem of displaying a space of several hundred dimensions would not have as satisfactory a solution.

This might be the case in a general pattern recognition problem, for which the features used for discrimination are much more varied than those used here. Thus, we believe that it is important to take advantage of the particular structure of speech in designing the data analysis tools. Specifically, the tools should strive to represent speech in the time-frequency space which is most familiar to speech scientists.

6.2.8 Diagnosing Model Deficiencies

In this section, we suggest possible extensions to the techniques presented thus far. While the extensions are generally applicable to recognizer error diagnosis, we will discuss them in the context of the case study to keep the exposition concrete.

We initially defined the error to be the misalignments between instances of "near" and the keyword's model. However, we have narrowed down the problem to be a phonetic confusion between /i^y/ and /y/ and will discuss error diagnosis in these terms. Before delving too deeply into the diagnosis, we should discard the possibility that the speaker actually realized the phoneme /i^y/ in "near" as a /y/. If this were the case, the problem would lie in the inability of the word model to account for such a pronunciation. By listening to some of these tokens and looking at their spectrograms, we discarded this possibility. Thus, the error is really due to a phonetic confusion.

We believe the problem to be a model deficiency rather then some sort of chance occurrence. This belief is reinforced by the fact that the confusion recurs for several tokens uttered by the same speaker. Moreover, the analyses of the previous few sections suggest that the errors are similar in structure and thus not due to chance. In the language of Tukey and Wilks, the model residuals are highly structured.

In fact, the relative score decomposition and segment/state profile suggest that the error can be attributed in large part to a single cause: the failure of the **iy** model to account for the possibility of an F_1 value as low as those observed for the segments in question. By identifying a single cause, we can simplify the diagnostic process by focusing on a low-dimensional subspace of the original measurement space. This simplification can be pushed to the limit by focusing on the measurement APF$_1$, the average predicted first formant, since it is most representative of the F_1 value throughout the segment. This measurement also contributes the most to the low relative score.

There are two possible causes of the mismatch between the observed APF_1 and its mean on state **iy3**: a deficient model of its probability distribution or an inability of the model to gener..ize from training data. We consider each in turn.

The PDF model might be poor because the discriminants are not truly distributed as a multivariate Gaussian. The multivariate Gaussian assumption implies that any linear combination of the discriminants has a Gaussian distribution [Johnson 88]. Thus, the back-transform $\boldsymbol{B}$ can be used to test the validity of the PDF model for an individual measurement such as APF_1 as follows: Let $\boldsymbol{Y}_{iy}$ be an $n_{iy} \times q$ data matrix of observation vectors used to train the state in question. Let $\boldsymbol{b}$ be the column of $\boldsymbol{B}$ and c be the element of $\mathbf{c}$ that are used to compute the least-squares estimate of APF_1. Then $\widehat{\mathbf{APF}}_{1,iy} = \boldsymbol{Y}_{iy}\boldsymbol{b} + c$ is the set of n_{iy} least-square-estimated APF_1 values for these vectors. The sample distribution of $\widehat{\mathbf{APF}}_{1,iy}$ can be used to compute a͏n empirical density estimate of $\widehat{\mathbf{APF}}_{1,iy}$. This in turn could be compared to the Gaussian distribution of APF_1 predicted from the state PDF parameters. The mean and stan^dard deviation of any back-transformed estimate's distribution are given by the two terms in Equation 6.41.

There exist several EDA techniques for comparing distributions, including smoothed histograms and normal probability plots [Chambers 83]. If this extension were to be implemented, a convenient user interface would be to allow these plots to be summoned by clicking on the relative score decomposition profile at the measurement of interest.

If the Gaussian PDF near the segments' estimated $\widehat{\mathrm{APF}}_1$ values were lower than the empirical density estimate, then the PDF model is likely responsible for the low segment score. Presumably, the estimated PDF of $\widehat{\mathrm{APF}}_1$ for a model that fit the training data better would be closer to the empirical density estimate. Thus, such a model would likely yield a higher score given the segment's $\widehat{\mathrm{APF}}_1$.

Note that we could compare the PDF's using the actual APF_1 values instead of their estimates. However, as we discussed in the derivation of the relative score decomposition, this would confound the effects of dimensionality reduction and PDF modelling. For the remainder of this section, we will assume, for the sake of clarity, that dimensionality reduction is not a problem. Thus, the diagnostic techniques to be discussed will use the least-squares APF_1 estimates and we will make no distinction between the APF_1 values and their estimates $\widehat{\mathrm{APF}}_1$.

If both the Gaussian PDF and empirical density estimate are very low near the segment APF_1 values, then the segment values are atypical of those seen in the training data. There are a variety of explanations for this. One is simply that there are not enough training data. Thus, the Gaussian model parameters may be poor estimates of the population parameters, i.e., those of the true distribution. This would account for the model's inability to generalize to previously unseen data. However, this is unlikely a problem in the present case because the state in question has was trained with over 1000 tokens.

Another possible explanation is that the PDF in the vicinity of the test segment APF_1 values would be higher if conditioned on factors aside from the phone label. For example, the observed APF_1 value might be more likely for tokens of /i^{y}/ that follow an /n/ than for all tokens of /i^{y}/ . This could be tested by computing the empirical PDF of all segments used to train state **iy3** that follow an /n/, assuming there were a sufficient number to make a good estimate of the density in the vicinity of the observed APF_1 value. If this density were higher than the unconditional one, then the use of a context-dependent model for /i^{y}/ would be suggested.[3]

It is more likely that speaker identity must be taken into account, since all low-scoring keyword tokens due to this speaker exhibit low APF_1 values. Thus, one promising line of investigation would be to look for evidence that the speaker has a relatively low APF_1 in other situations. If this were the case,

[3] Of course, it is known that such models improve performance in general and so it can be argued that we do not need to use EDA techniques to suggest such improvements. We discuss this issue in more detail in the next section.

then failure to adapt to the speaker might be the key model deficiency.

While it is unlikely the case here, another type of deficiency might be the failure to utilize measurements useful for phonetic discrimination. This problem is akin to that of too much dimensionality reduction. With too much dimensionality reduction, information useful for discrimination is removed from the measurement set. However, transforming the waveform into the measurement set is itself a form of dimensionality reduction. Brown [Brown 87] pointed out this similarity as well. To diagnose this problem, it is necessary to identify features of the original waveform that would be useful for distinguishing /i^y/ from /y/ and that are not represented in the measurement set. Such a diagnosis must rest on the system designer's prior acoustic-phonetic knowledge.

The segmentation process might be responsible for loss of information as well. This is possible regardless of whether the segments are of variable or fixed duration. With variable-duration segments, a missed phonetic boundary that does not match a biphone model should be easy to identify from the segment-score plot, since it includes a spectrogram. In a frame-based system, a possible problem is the inclusion in a single frame of a sudden event such as a stop burst with a very different acoustic event such as voicing, thus making identification of the stop difficult. In fact, Brown [Brown 87] noticed this phenomenon. Again, the segment-score plot should be useful in revealing this problem.

6.2.9 Summary of Speech Recognizer Diagnostics

Table 6.2 summarizes many of the techniques introduced in the last two chapters for diagnosing spotter errors. Items ending in question marks relate to techniques that were suggested but not implemented.

Detail	Representation	User interface	Case study findings
Overall performance	Area over ROC	Keyboard	Phone-trained models superior
Individual trial results	Trial score scatter plot	Keyboard	Low scores on speaker "reg"
Segment scores	Segment score plot	Click on trial score	Misaligned tokens
Forced alignment	Forced segment score plot	Click on segment score plot	**/n/-ix-n**, /i^y/ **-y** confusions
Relative score decomposition	Relative score profile	"	Confusion related to PF_1 values
Measurement vector vs. model means	Segment-state profile	"	Confusion due to low PF_1
Gaussian vs. empirical PDF	Smoothed histogram?	Click on relative score profile	Underestimated APF_1 PDF near segment value?
Empirical PDF vs. segment score	"	?	Insufficient training, unmodelled factors?

Table 6.2: Summary of special-purpose EDA techniques. Question marks denote techniques suggested but not implemented.

6.2.10 Extending the Techniques

While we have demonstrated these EDA techniques in the context of word spotting, they could easily be adapted to analyzing the behavior of a continuous speech recognizer. Words or subwords assigned particularly low relative scores by the recognizer could be flagged and represented in a plot analogous to the trial score scatter plot. The segment score plot could be applied exactly as is in the word spotter, comparing scores on the correct word model and its closest competitor.

The relative score decomposition and segment-state profiles can be used in their current form by any recognizer that uses multivariate Gaussian representations. Neither of these techniques relies on the fact that the state PDF's are modelled with diagonal covariance matrices. Nor do they rely on the fact that the transformation from the measurement to the observation space uses principal components or discriminant analysis. Thus, for example, the techniques could be used in a frame-based cepstrum-based recognizer which uses a full covariance matrix to model the PDF to diagnose errors in the spectral domain, since the cepstrum is a linear transformation of the log spectral energies.

However, the score decomposition does rely on the fact that the log observation probability can be decomposed into a sum of terms in the measurement space. This is not true of a recognizer that uses mixture Gaussian PDF models since the log observation probability or segment score λ for a state in such a recognizer is of the form

$$\lambda = \log \sum_{i=1}^{m} u_i f_i(\boldsymbol{y}) \tag{6.44}$$

where m is the number of mixtures, u_i is the weight on each mixture, and $f_i(\boldsymbol{y})$ is the PDF of the i^{th} mixture evaluated at the observation vector $\boldsymbol{y}$. We are confident that even if there is no closed form solution to the problem of applying score decomposition to the mixture case, good heuristics can be found for doing so. We believe that this is a worthwhile avenue of research given the recent evolution towards the use of mixture models in speech recognizers.

The techniques introduced in this section are only a sample of the possible uses of EDA in speech recognizer design. There are sure to be situations in which other special purpose techniques can be used to make specific types of diagnoses. More important than the techniques themselves is the philosophy behind them: namely that exploring the data can lead to the generation of hypotheses that would not have been made otherwise and these, in turn, can lead to building better models. We discuss this further in the next section.

6.3 How can EDA be Used to Improve State-of-the-Art Speech Recognizers?

6.3.1 Applying the Strategy

The system designer's ultimate goal, of course, is not to diagnose speech recognizers but to "cure" them of their errors and we have yet to show directly how EDA can be used to accomplish this. Until now, in fact, EDA has played little, if any, such role. The process employed to improve recognizers has been to:

1. build acoustic models,
2. measure an overall error rate,
3. deal with perceived problems in the models based on general knowledge about statistical modelling and speech by changing all the models in a general way.

At this point, the cycle repeats. Changes in the models that lead to error rate reductions are preserved and others are not.

In this manner, HMM recognizers have progressed from phone models based only on cepstral measurements and a single VQ codebook [Rabiner 83] to the current state of the art. The currently best-performing recognizers often include generalized triphones, cepstral difference measurements, tied-mixture continuous density representations, gender-specific models and perhaps discriminative training, e.g., [Cohen 90]. Published reports of the breakthroughs made along the way do not attribute motivations to intensive error analysis but rather to reasoning based on general principles. For example, triphones are a response to the problem of coarticulation, cepstral differences to the fact that spectral changes are often cues to phonetic identity, and gender-specific models to known differences in speech between the genders. Mixtures and discriminative training, on the other hand can be justified on statistical modelling grounds. Even without looking at errors, it can be reasoned that vector quantization introduces distortion while single-mode multivariate Gaussian models are not robust to deviations from the Gaussian assumption [Bellegarda 90], and that maximum likelihood approaches are suboptimal when the correct family of PDF models is unknown [Brown 87].

However, even state-of-the-art recognizers come nowhere close to attaining human performance. Thus, the models are still deficient. There are two basic strategies for improving the models and their relative effectiveness depends on the structure of the errors made by recognizers. The first possibility is that the errors have little structure, i.e., they are distributed evenly across different speakers, types of phonetic confusion, and other factors, and cannot be blamed on gross deviations from the assumed probability model. If such is the case, then a few specific modelling improvements would likely have little effect. The best strategy in this case would probably be to build more detailed PDF models and to collect more training data for making reliable parameter estimates of these models. We term these *quantitative* model improvements.

If, on the other hand, the errors are structured, a few specific modelling improvements should be effective in reducing their number. The ideas for these improvements might continue to be derived solely from general principles. These ideas could be tested according to the paradigm outlined above, using overall error rate to gauge their utility. However, we believe that a more effective strategy would include the following steps:

1. Use EDA to diagnose errors on some test data, as in the case study of Section 6.2.

2. Form hypotheses about the model deficiencies responsible for the errors based on general principles, e.g., speaker-dependence of F_1.

3. Perhaps use EDA to test the validity of these hypotheses on the same test data set. For example, in the case study, we could see if the speaker's F_1 is relatively low compared to phone model predictions on tokens and phone models aside from those initially diagnosed.

4. Devise models to remedy these deficiencies. For example, utilize a normalization or adaptation technique to cause the model's prediction of the speaker F_1 to be closer to the realized one.

5. Test the models on the test data to see if they remove the errors originally diagnosed and also monitor changes in overall performance to see if the remedy has unintended side effects that reduce overall performance. If there are side effects then perhaps they can be cured by some other method. If not, then the particular modification to the models should not be preserved.

6. Repeat the previous step on new data, monitoring both overall performance and errors which the new models were designed to cure. This is an important step for ensuring that the phenomena identified on the original test set are general, and not quirks specific to the test set. It is far more important to do this when using an EDA approach than when using overall performance measures. Otherwise, there is a real danger of the approach degenerating into a "rule-based" system, with lots of model changes that only work on the original test data.

By targeting errors actually made by the recognizer instead of relying on general principles to generate hypotheses of model deficiencies, we believe that faster progress can be made in improving performance, particularly as we exhaust the hypotheses suggested by general principles. In other words, EDA can fulfill the goal suggested by Tukey and Wilks of "laying open the data to display the unanticipated," where in this case the deficiencies are unanticipated based on general principles.

In a sense, the approach of targeting errors is analogous to automatic discriminant techniques which iteratively alter models in response to errors made by them. However, there is an essential difference: discriminant techniques and automatic techniques in general can make quantitative model improvements by changing model parameters in response to errors but there is no evidence that they can effect qualitative changes in the models, such as the incorporation of speaker adaptive models where none existed or the inclusion of new measurements believed to be more invariant across different speakers, phonetic contexts, or environments.

In our view, it is in making these qualitative changes that human-mediated methods such as EDA hold the most promise compared to automated techniques. This distinguishes them from other human-mediated model building techniques such as rule-based systems. Such techniques rely on human intervention for both qualitative improvements *and* quantitative model improvements such as parameter setting. We believe the latter is better left to automated techniques.

The findings of the case studies in this chapter and in Chapter 5 lead us to believe that errors made by speech recognizers are structured and thus can be cured by qualitative model changes. In both cases, spotter errors were shown to be caused by one or two phenomena. Moreover, in the case of the poor-performing single-pronunciation networks, we were able to eliminate the error by using a less restrictive word model. Although we made this improvement based on general principles rather then data analysis and only performed the data analysis after the fact, it is not hard to imagine the reverse sequence of events. While these are only two cases, they were the only ones we have looked at in any depth with data analytic tools, not ones we chose to discuss because they were particularly interesting. Thus, we predict that if we looked at other errors as intensively we would find structure as well.

6.3.2 Special- vs. General-Purpose Modelling Techniques

Techniques used to make qualitative model changes can be classified as special- or general-purpose. Special-purpose techniques devote specialized models to particular phenomena, thus *leading to a heterogeneity of models in the recognizer*. A good example is function-word modelling [Lee 88]. Such techniques are useful when a few phenomena are responsible for a large number of errors (for example, function word confusions), and are also feasible to use as the chief methodology when building recognizers specialized to a single task (for example, the digit recognizer of [Bush 87]). EDA can be particularly valuable in suggesting heterogeneous techniques since it can be used to identify a small subset of phenomena requiring special models. In fact, Bush and Kopec [Bush 87] made substantial use of EDA techniques in building a high-performance digit recognizer.

Special-purpose techniques can be implemented in an otherwise homogeneous recognizer, as the function-word models are. A second way of implementing them is in a two-pass system, in which the homogeneous recognizer passes high-scoring hypotheses to special-purpose models for rescoring. The two-pass paradigm has been advocated recently by Schwartz [Schwartz 92] to be used in conjunction with N-best search procedures and has been shown to lead to improved performance in cases where a computationally cheap technique is used in the first pass and a more expensive one used in the second [Austin 92]. While these applications tend to use homogeneous models in both passes, we can envision several profitable applications of a special-purpose second pass. For instance, one model deficiency that might be identifiable by EDA techniques is loss of information due to dimensionality reduction. Remedying this by using a longer observation vector might cause overall performance to drop because of insufficient training data. However, the longer vector might be successful in remedying certain types of error in a second pass as long as the correct model can score high enough to be considered.

General-purpose techniques are applied to the complete set of acoustic models. Examples include mixture models and triphone models. They are useful in correcting systemic model deficiencies. For example, mixture models account for deviations from normality and triphone models account for coarticulation, which is not modelled well with context-independent models. EDA can be used to diagnose such deficiencies. In our case study, for example, it is quite possible that speaker adaptation or normalization would combat the observed errors due to speaker "reg" and presumably would be useful in general. By focusing on errors whose likely cause is failure to account for speaker dependence and by determining the structure of these errors, methods for dealing with the deficiencies might be developed that would be hard to conceive of otherwise.

In the best possible case, the methods would lead to more parsimonious models. These in turn could lead to both greater robustness and reduced computation. For instance, while increasing the number of mixture components in a recognizer can lead to increased performance and mixture models can in the limit be fit to any distribution [Ney 90], more components require more computation and more training data. The same things are true if triphone models instead of phone models are used. Some of the burden could be reduced if particular model deficiencies could be identified more precisely and dealt with more efficiently. For instance, mixtures probably acccunt for speaker variation, contextual variability, and miscellaneous outliers in the training data better than do simpler models. If these factors could be extricated from each other through EDA, each of them might be able to be dealt with specifically, perhaps leading to simpler models. Again, the role of EDA would be to not only diagnose deficiencies but to prompt cures that might not otherwise be considered.

Note that we have advocated two contradictory approaches to improving recognizers. The special-purpose techniques tend to increase complexity while the general purpose ones can potentially decrease it through the development of more parsimonious models. We feel that both approaches have a role to play. Parsimonious solutions are better when they can be found and implemented in the existing system since they can lead to simplification. However, if such solutions cannot be found, heterogeneous ones may be called for.

6.3.3 The Costs of EDA

The benefits of using EDA must be weighed against the time it takes to apply it, since it is obviously a labor-intensive methodology. Speech recognizers usually include many parameters and their design involves the use of very large datasets. Unless effective methods are used for focusing on important details and processing data rapidly, the costs will outweigh the benefits. We believe that the hierarchical organization of plots introduced in Section 6.2 is a useful paradigm in this regard, since each level highlights possible areas of model deficiency that can be investigated at the a more detailed level with a mouse click. For very large data sets, automatic techniques could be used to find tokens which are above a certain "importance threshold" (for example those with very low relative scores). Alternatively, small random samples of a large data set could be studied intensively. Another key point in making the analysis tractable is to reduce the dimensionality of the phenomena being studied, which was a goal of the score decomposition. Dimensionality reduction has also been suggested in [Good 83] as an important EDA technique.

A second cost is the time required to implement the analysis tools. Software packages such as S-Plus designed especially for data analysis make the implementation more feasible. All the plots introduced in Section 6.2, including the spectrogram, were composed of high-level S-Plus plotting routines. Additionally, the S language [Becker 88], upon which S-Plus is based, has a fairly rich set of data structures and is fairly flexible. Thus, the data manipulations required for implementing the techniques were straightforward to write. While we carried the use of S to an extreme ana wrote almost all the code for training and recognition in it, this is not necessary since facilities exist for importing binary data into S-Plus. There may be other packages convenient for implementing these techniques as well.

We end our discussion of the possible drawbacks of using EDA by warning that there is always a danger of jumping to unwarranted conclusions prematurely. A peculiarity in the data being analyzed might be confused with a real effect, especially if the observed phenomenon is in accord with one's preconceived ideas about the data. Diaconis [Diaconis 85] calls this "magical thinking," which he defines as the "inclination to seek and interpret connections between the events around us, together with our disinclination to revise belief after further observation." It is to avoid this phenomenon that we suggested above the importance of testing models on previously unseen data. Diaconis offers this, and other suggestions as well, for avoiding the problem. This solution may be costly in terms of the amount of data needed to practice EDA. Again, the costs must be weighed against the benefits.

6.4 Summary

In this chapter, we have introduced the methodology of exploratory data analysis, related it to some of the work of previous chapters, and introduced techniques designed for applying EDA to speech recognizer development. Finally, we discussed the role that EDA can play in improving the performance of state of the art recognizers and advocated a paradigm in which it plays a much greater role in system development than it has till now.

The contributions of this chapter are both philosophical and technical. The major philosophical contribution is the framing of speech recognizer design as a statistical modelling process in which we iteratively fit models to data, look for structure in the residuals of the fit using EDA, and try to account for that structure by improving the models. The description is drawn from that used by Tukey and Wilks in their paper about the philosophical underpinnings of EDA [Tukey 86]. In the case of speech recognition, we have equated the residuals to recognizer errors. We used this framework to guide the development of the techniques introduced in Section 6.2 and our suggestions in Section 6.3 about how EDA can be applied to speech recognition development most profitably. As we stated in that section, one of the key contributions that EDA can make along these lines is to reveal model deficiencies unanticipated by the system designer.

The major technical contributions have been the development of techniques for diagnosing speech recognition errors at a detailed level and the application of these techniques to a case study of the word spotter. The techniques make use of graphical representations and linear transformations to display the data in a form interpretable by the system designer.

The key findings in the case study were (a) that most of the low-scoring keyword tokens in the spotter were due to a single speaker, and (b) that the spotter's poor performance on each of these tokens could be attributed to a small number of measurements whose values were far from that expected by the keyword model. These findings, along with those of Chapter 5 concerning the single-pronunciation model for "MIT", encourage us to believe that in general, it will be possible to identify a small number of deficiencies that account for a disproportionate number of recognizer errors. By remedying the deficiencies, it should be possible improve recognizer performance. Thus, the effort spent in error diagnosis will likely be repaid in greater performance.

Chapter 7

Summary, Future Work, and Conclusions

7.1 Summary and Conclusions

We now summarize previous chapters briefly, emphasizing the main findings. More detailed summaries, discussions and suggestions for future work appear at the ends of each chapter.

In Chapter 3, we introduced the segment-based HMM. Like the conventional frame-based HMM, it consists of two stages. In the first stage, the incoming signal is split into segments that do not overlap in time. In the second stage, measurements are made on these segments. The frame-based and segment-based systems differ in that the segments in a frame-based HMM each have a fixed duration while those of the segment-based HMM are determined by a an algorithm that determines segment boundaries based on acoustic criteria. The segmenter attempts to delineate regions that are associated with distinct phones. Such a strategy has three potential advantages over frame-based HMM's:

1. The statistical dependence of short-time spectral measurements made near each other can be modelled better.

2. The segmenter tends to place boundaries at points of large spectral change, thus identifying points which are believed to be rich in infor-

mation useful for phonetic discrimination.

3. Such a scheme produces fewer segments per unit time than is typical in frame-based systems, leading to computational savings.

The scheme also has theoretical and practical advantages over the stochastic segment approach [Ostendorf 89, Zue 89a]. Theoretically, HMM scoring is better-formulated mathematically than stochastic segment scoring, as we pointed out in Section 1.3. Practically, the scheme can be computationally cheaper if it uses a fast segmenter and separates segmentation from recognition.

We showed that the segmenter behavior is systematic. Thus, it could potentially be well-modelled within the HMM framework. For instance, phonetic regions over which there was a large degree of spectral change, e.g., diphthongs, tended to be associated with more segments than regions that consisted of a single short event, such as those of voiced stops. An HMM that could match segment sequences of various lengths was used to model this variability.

On the other hand, acoustically similar or coarticulated phonetic regions were often merged into a single segment or into a sequence of segments whose boundaries differed from those of the phonetic regions. To account for this phenomenon, we employed biphone models, which modelled most of the merged phonetic regions. Because there were relatively few tokens available to train these models, we pointed out in Chapter 3 that their use might represent a weakness of the segment-based HMM approach. In Chapter 4, we showed that this was not the case by analyzing the results of phonetic recognition experiments.

However, accounting for all possible biphones or even for all those that were merged at least once in the training set would have required too many models, leading to infeasibly large computational requirements and very small training sets for some of the biphone models. Thus, we accounted for only those that had the most training data associated with them. As we pointed out in Chapter 3, in a large vocabulary recognizer some mechanism would have to be devised to account for all the merged labels. The need to account for all merges might be a shortcoming of the approach.

Chapter 4 had three goals. The first was to investigate the effectiveness of different measurement sets for acoustic modelling of segments. We used a phonetic recognition task to gauge their relative effectiveness. We were also interested in investigating how acoustic-phonetic knowledge is represented in these measurements and to use the results of these investigations to improve phonetic recognition performance. The third goal of the chapter was to compare phonetic recognition performance of the segment-based HMM to that of existing approaches.

The chief finding of of the measurement set comparison was that the addition of spectral measurements made 5 and 35 ms before and after the boundaries of the modelled segment to those made within the segment could lead to a substantial and statistically significant increase in performance. Conversely, there was little gain when spectral representations of the beginning and end of the segment were added to a representation of the segment's center. We attribute the latter finding to the fact that the segment-based HMM can account for spectral change within a phonetic region without the need for spectral measurements to be made at multiple locations within a segment.

We were quite successful in developing insight into knowledge representation but less so in applying the results to phonetic recognition. For instance, we were able to show that linear combinations of mel-frequency spectral coefficients modelled formants poorly. We built a multiple regression model based on nonlinear transformations of the coefficients that performed much better at modelling the formants, albeit over a limited frequency range. However, we were unable to significantly improve phonetic recognition performance by including these formant estimates in the acoustic measurement set. In another case, we chose the positions to make the out-of-segment measurements discussed in the previous paragraph so as to minimize the number of confusions among stop consonants. However, while the addition of out-of-segment measurements resulted in a large overall reduction in the number of phonetic misclassifications, there was little reduction in the number of confusions among stops.

We thus believe that to reap the benefits of incorporating acoustic-phonetic knowledge in a recognizer, it is important to analyze recognizer behavior in detail, focusing on the effect of modelling assumptions that intervene between the measurement set and the recognizer output. By understanding the effect of these assumptions, models that better exploit the addition of knowledge can potentially be built. The development of techniques for conducting this detailed analysis was the main subject of Chapter 6, to be summarized below.

Our comparison of phonetic recognition performance of the segment-based HMM to that of existing approaches provided evidence that the segment-based HMM is competitive with the others. A direct comparison is hampered by the fact that the corpora used to test the systems investigated here and elsewhere are different. Also, we tested our system on male speakers only while previous experiments concerned speakers of both genders. However, we believe that these differences did not have a major effect on our results. Thus, we conclude that the segment-based HMM approach is promising enough to pursue further by refining aspects of the system.

We applied the segment-based HMM to word modelling in Chapter 5. The main goals of that chapter were to investigate certain issues pertinent to HMM word modelling in general and in the segment-based HMM framework in particular and to introduce the segment score plot, a technique for diagnosing recognizer errors. For each word in our study, we compared different word models by evaluating the performance of a spotter for that word as a function of various models of the word. The models varied along three dimensions: training method, type of pronunciation network, and measurement set.

In the course of this investigation, we developed novel algorithms for word spotter scoring and performance evaluation. The algorithms are particularly suited to detailed error analysis, one of the main topics of the thesis, but have certain features that are generally appealing for word spotting implementations.

The models that performed the best overall were trained from word-specific data and employed networks intermediate in bushiness between those that allow single pronunciations and those that allow many alternate pronunciations. These results held for both function and content words. We conclude from this result that even with a small number of training tokens, word-specific models can outperform subword-based models.

Also, we noted that when subword-based models were used, the bushiest pronunciation networks we employed tended to perform best. The interaction between training method and network type should be kept in mind in applications where it might be infeasible to train word-specific models for every word in the vocabulary.

Single-pronunciation networks performed particularly badly, a result we studied more closely using the segment score plot. We were able to demonstrate that for the word "MIT", the single pronunciation network failed because it was unable to account for variability in the realization of stop closures. We concluded from this case study that detailed error analysis techniques could be useful for discovering model deficiencies.

Finally, we found that the effect of measurement set on performance was similar to that found in the phonetic recognition task of Chapter 4: a measurement set that included out-of-segment measurements outperformed one that did not by a wide margin for almost all words in the study.

In Chapter 6, we introduced the principles of exploratory data analysis and showed how they could be applied to the design of speech recognition systems. In particular, we advocated an iterative approach involving model building, error analysis, and model improvement.

We developed interactive graphical techniques for performing detailed error analysis and applied them to a case study of the spotter for the word "near." The techniques are organized hierarchically so that a system designer could investigate errors at various levels of detail. At the most detailed level, we developed a technique for decomposing acoustic scores for a segment into subscores associated with distinct spectral measurements. We also developed a format for displaying the differences in subscores between the actual and hypothesized segment label in a manner such that they could be easily interpreted.

We showed that for the case study, the tokens of "near" which were scored lowest by the word spotter were likely to be uttered by a particular speaker. Furthermore, all such tokens received low scores because F_1 throughout the fronted vowel in "near" was very low compared to the value expected by the "near" model. We infer from this finding and from the finding in Chapter 5 concerning "MIT" that a disproportionate number of recognizer errors might often be able to bc attributed to a small number of causes. If so, then a small number of model improvements could potentially lead to a large performance improvement in a speech recognition system. Error diagnostic tools are important in this regard in that they allow the identification of modelling deficiencies that must be dealt with to effect such an improvement. While we did not use the tools to improve performance in this thesis, we suggested strategies for doing so.

7.2 Future Work

Several areas of this work are worthy of further investigation. First of all, to better evaluate the segment-based HMM approach, it should be tested on the same phonetic recognition task as that used by other researchers, using a large set of speakers from the TIMIT corpus. As we have mentioned, there are several ways in which we believe the system can be improved. In particular, techniques such as context-dependent [Lee 90] and and tied-mixture Gaussian PDF modelling [Bellegarda 90] that have been shown to improve frame-based

HMM performance should improve segment-based HMM performance as well. Also, as we pointed out in Chapters 3 and 4, alternate segmentation algorithms, HMM topologies, methods for associating segments with labels, and strategies for choosing the subword label inventory should be explored. None of these aspects of the system were optimized based on phonetic recognition or word spotting performance. Work along these lines should lead to improved system performance.

Clearly, there is more work to be done in the acoustic modelling of segments, as well, particularly in the realm of employing measurements based on acoustic-phonetic knowledge. As we suggested in the discussion of formant modelling in Chapter 4, one strategy for choosing such measurements is to determine non-linear transformations of spectral measurements for modelling attributes of speech believed to be important for discrimination, such as formant frequencies and voice onset time. To get the most out of adding these knowledge-based measurements, it will be desirable to analyze in detail their effect on performance using tools similar to those developed in Chapter 6.

The problem of word modelling, particularly in the segment-based HMM framework, should also be investigated further. The choice of pronunciation network, in particular, seems to have a large effect on word model performance. More general and principled methods for determining word networks than those we employed should be developed. To apply segment-based HMM's to continuous speech recognition, there are problems that remain to be worked out. For example, biphones can occur across word boundaries, as occurred in the case study of Chapter 6, for exa.nple. This introduces complexities in modelling the beginning and ending of words. In some cases, word pairs that frequently share a biphone might have to be modelled as a single unit.

Finally, much remains to be done in the application of exploratory data analysis to speech recognizer design. The most important extension, of course is to show that the techniques can actually be used to improve recognizer performance, rather then simply diagnosing errors. In Chapter 6, we suggested several general methods for making such improvements (e.g., two-pass methods based on the N-best search [Schwartz 92]) based on the findings of detailed error analyses. We also showed that, for the cases we studied at least, errors tend to be highly structured and therefore "curable" by making a relatively small number of modelling improvements. Thus, we are confident that the data analytic methodology can be used profitably. However, to be convinced of this, we would prefer to apply it in the development of a practical speech recognition application.

7.3 Concluding Remarks

Throughout our work, we have investigated the problem of representing acoustic-phonetic knowledge in statistical models. We chose the segment-based HMM for our work in large part because we believe that measurements made on segments are superior to those made on frames for representing such knowledge. At the same time we retained the HMM framework because it has been shown to be a powerful formalism for statistical modelling of speech.

In the course of this investigation, we conducted a series of experiments in which well-known statistical techniques - multiple regression, clustering, discriminant analysis - were used to explore the relationship between the set of measurements used to characterize segments and attributes known to be relevant for phonetic discrimination, such as formants and distinctive features. The experiments were also designed to test whether measurements and transformations developed with such techniques could be used to improve recognizer performance.

Finally, we developed tools for analyzing recognizer behavior and diagnosing errors. The tools are aimed at enabling the system designer to visualize the complicated statistical processes underlying recognizer behavior in terms of his/her acoustic-phonetic knowledge base. Thus, they can be used to identify deficiencies in the modelling process that intervene between the input of acoustic-phonetic knowledge and the recognizer output. Once the deficiencies are identified, it should become clearer to the designer what steps to take to achieve better recognition performance.

As we pointed out in the thesis introduction and in Chapter 6, the emphasis on knowledge representation and detailed analysis is not shared by all researchers in the field of automatic speech recognition. Instead, the prevailing approach emphasizes more sophisticated statistical approaches, more training data, and more computation. It is difficult to refute claims that the prevailing approach will lead to inexorable improvement in the state of the art of speech recognition. However, we believe that as recognizers come to be applied more frequently in real-world problems in which humans outperform automatic recognizers, e.g., noisy environments, multiple speakers, and spontaneous speech, it will be necessary to not only better understand the structure of speech but to apply that understanding to build better statistical models for speech recognition. We hope that the model building philosophy and the specific methods we have introduced will facilitate bridging the gap between speech knowledge and speech recognizer design.

Bibliography

[André-Obrecht 88] André-Obrecht, R., "A New Statistical Approach for the Automatic Segmentation of Speech Signals," *IEEE Trans. on ASSP*, Vol. 36, 29-40, January, 1988.

[Atal 71] Atal, B. S. and Hanauer, S. L., "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *JASA*, Vol. 50, 637-655, 1971.

[Austin 92] Austin, S., Zavaliagkos, G., Makhoul, J., and Schwartz, R. "Speech Recognition Using Segmental Neural Nets," *Proc. ICASSP 92*, I 625-628, March, 1992.

[Bahl 80] Bahl, L. R., Bakis, R., Cohen, P. S., Cole, A., Jelinek, F., Lewis, B. L., and Mercer, R. L., "Further Results on the Recognition of a Continuously Read Natural Corpus," *Proc. ICASSP 80*, 872-875, April, 1980.

[Bahl 81] Bahl, L. R., Bakis, R., Cohen, P. S., Cole, A., Jelinek, F., Lewis, B. L., and Mercer, R. L., "Speech Recognition of a Natural Text Read as Isolated Words," *Proc. ICASSP 81*, 1168-1171, May, 1981.

[Bahl 83] Bahl, L. R., Jelinek, F., and Mercer, R. L., "A Maximum Likelihooa Approach to Speech Recognition," *IEEE Trans. on PAMI*, Vol. PAMI-5, 179-190, March, 1983.

[Becker 88] Becker, R. A., Chambers, J. M., and Wilks, A. R., *The New S Language*, (Pacific Grove, CA: Wadsworth & Brooks/Cole, 1988.)

[Bellegarda 89] Bellegarda, J. R. and Nahamoo, D., "Tied Mixture Continuous Parameter Models for Large Vocabulary Isolated Speech Recognition," *Proc. ICASSP 89*, 13-16, May, 1989.

[Bellegarda 90] Bellegarda, J., R. and Nahamoo, D., "Tied Mixture Continuous Parameter Models for Speech Recognition," *IEEE Trans. on ASSP*, Vol. ASSP-38, 2033-2045, December, 1990.

[Bellegarda 92] Bellegarda, J. R., de Souza,, P. V., Nádas, A. J., Nahamoo, D., Picheny ,M. A., and Bahl, L. R., "Robust Speaker Adaptation Using a Piecewise Linear Acoustic Mapping," *Proc. ICASSP 92*, I 445-448, March, 1992.

[Bocchieri 86] Bocchieri, E. L. and Doddington, G R., "Frame-Specific Statistical Features for Speaker Independent Speech Recognition," *IEEE Trans. on ASSP*, Vol. ASSP-34, 755-764, August, 1986.

[Broad 89] Broad, D. J. and Clermont F.. "Formant Estimation by Linear Transformation of the LPC Cepstrum," *JASA*, Vol. 86, 2013-2017, November, 1989.

[Brown 87] Brown, P. F., "The Acoustic Modelling Problem in Automatic Speech Recognition," Ph. D. Thesis, Computer Science Department, Carnegie Mellon University, May, 1987.

[Bush 87] Bush, M. A. and Kopec, G E., "Network-Based Connected Digit Recognition," *IEEE Trans. on ASSP*, Vol. ASSP-35, 1401-1413, October, 1987.

[Carlson 75] Carlson, R., Fant, G. and Granstrom, B., "Two-Formant Models, Pitch and Vowel Perception," in *Auditory Analysis and Perception of Speech*, G. Fant and M. A. A. Tatham, (eds.), 55-82, (London: Academic, 1975.)

[Chambers 83] Chambers, J. M., Cleveland, W. S., Kleiner, B., and Tukey, P. A. *Graphical Methods for Data Analysis.* (Belmont, CA: Wadsworth International Group and Boston, MA: Duxbury Press, 1983.)

[Chigier 92] Chigier, B, "Rejection and Keyword Spotting Algorithms for a Directory Assistance City Name Recognition Application," *Proc. ICASSP 92*, II 93-96, March, 1992.

[Chow 86] Chow, Y., Schwartz, R., Roucos, S., Kimball, O., Price, P., Kubala, F., Dunham, M.O., Krasner, M. and Makhoul, J., "Context-Dependent Modelling for Acoustic-Phonetic Recognition of Continuous Speech," *Proc. ICASSP 86*, 1593-1596, April, 1986.

[Chow 87] Chow, Y. L., Dunham, M. O., Kimball, O. A., Krasner, M. A., Kubala, G. F., Makhoul, J., Roucos, S., Schwartz, R. M., "BYBLOS: The BBN Continuous Speech Recognition System," *Proc. ICASSP 87*, 89-92, April, 1987.

[Cohen 74] Cohen, P. S. and Mercer, R. L., "The Phonological Component of an Automatic Speech Recognition System," IEEE Symposium on Speech Recognition, Carnegie-Mellon University, 177-187, April, 1974.

[Cohen 90] Cohen, M., Murveit, H., Bernstein J., Price P., and Weintraub, M. "The DECIPHER Speech Recognition System," *Proc. ICASSP 90*, 77-80, April, 1990.

[Cole 86] Cole, R., Stern, R. M., and Lasry, M. J., "Performing Fine Phonetic Distinctions: Templates versus Features," in *Invariance and Variability in Speech Processes*, J. S. Perkell and D. H. Klatt, (eds.), 325-342, (Hillsdale, NJ: Lawrence Erlbaum Assoc., 1986.)

[Colla 85] Colla, A. M., Scagiola, C. and Sciarra, D., "A Connected Speech Recognition System Using a Diphone-Based Language Model," *Proc. ICASSP 85*, 1229-1232, August, 1985.

[Colla 86] Colla, A. M., "Some Considerations on the Definition of Sub-Word Units For a Template-Matching Speech Recognition System," *Proceedings of the Montreal Symposium of Speech Recognition*, 55-58, July, 1986.

[Cyphers 86] Cyphers, D. S., Kassel, R. H., Kaufman, D. H., Leung, H. C., Randolph, M. A., Seneff, S., Unverferth, J. E. III, Wilson, T., and Zue, V. W., "The Development of Speech Research Tools on MIT's LISP Machine-Based Workstations," *Proc. Speech Recognition Workshop*, Palo Alto, CA, February 19-20, 1986.

[Davis 80] Davis, S. B. and Mermelstein, P., "Comparison of Parametric Representations of Monosyllabic Word Recognition in Continously Spoken Sentences," *IEEE Trans. on ASSP*, Vol. ASSP-28, 357-366, August, 1980.

[DeLattre 55] DeLattre, P. C., Liberman, A. M. and Cooper, F. S., "Acoustic Loci and Transitional Cues for Consonants," *JASA*, Vol. 27, 769-773, July, 1955.

[Delong 88] Delong, E.R., Delong, D.M., and Clarke-Pearson, D.L., "Comparing the Areas Under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach," *Biometrics* **44**, 837-845, September, 1988.

[Deng 88] Deng, L., Lennig M., Gupta, V. N. and Mermelstein, P., "Modelling Acoustic-phonetic Detail in an HMM-based Large Vocabulary Speech Recognizer," *Proc. ICASSP 88*, 509-512, September, 1988.

[Deng 89] Deng, L., Lennig, M., and Mermelstein, P., "Use of Vowel Duration Information in a Large Vocabulary Word Recognizer," *JASA*, Vol. 86, 540-548, August, 1989.

[Deroualt 88] Deroualt, A.-M., "Context-dependent Phonetic Markov Models for Large Vocabulary Speech Recognition," *Proc. ICASSP 88*, 509-512, September, 1988.

[Diaconis 85] Diaconis, P., "Theories of Data Analysis: From Magical Thinking Through Classical Statistics," in *Exploring Data Tables, Trneds and Shapes*, Hoaglin, D. C., Mosteller, F., and Tukey, J. W. (eds.), 1-36 (New York, NY: John Wiley & Sons, Inc., 1985.)

[Digilakis 92] Digilakis, V. V., "Segment-Based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition," Ph. D. Thesis, Boston University Graduate School, 1992.

[Dillon 84] Dillon, W. R. and Goldstein M., *Multivariate Analysis.* (New York, NY: John Wiley & Sons Inc., 1984.)

[Doddington 89] Doddington, G., "Phonetically Sensitive Discriminants for Improved Speech Recognition," *Proc. ICASSP 89*, 556-559, May, 1989.

[Duda 73] Duda, R. O. and Hart, P. E., *Pattern Classification and Scene Analysis.* (New York, NY: John Wiley & Sons Inc., 1973.)

[Espy-Wilson 87] Espy-Wilson, C. Y., "An Acoustic-Phonetic Approach to Speech Recognition: Application to the Semivowels," Ph. D Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, December, 1987.

[Fisher 86] Fisher, W. M., Doddington G. R. and Goudie-Marshall, K. M., "The DARPA Speech Recognition Research Database: Specifications and Status," *Proc. DARPA Speech Recognition Workshop,* Report No. SAIC-86/1546, 93-97, 1986.

[Fissore 91] Fissore, L., Laface,P. and Micca, G., "Comparison of Discrete and Continuous HMM's in a CSR Task over the Telephone," *Proc. ICASSP 91*, 253-256, May, 1991.

[Furui 86] Furui, S., "Speaker-Independent Isolated Word Recognition using Dynamic Features of Speech Spectrum," *IEEE Trans. on ASSP*, Vol. ASSP-34, 52-59, February, 1986.

[Gillick 89] Gillick, L. and Cox, S., "Some Statistical Issues in the Comparison of Speech Recognition Algorithms" *Proc. ICASSP 89*, 532-535, May, 1989.

[Glass 84] Glass, J. R., "Nasal Consonants and Nasalized Vowels: An Acoustic Study and Recognition Experiment," S. M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, December, 1984.

[Glass 88] Glass, J. R., "Finding Acoustic Regularities in Speech: Applications to Phonetic Recognition," Ph. D Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, December, 1988.

[Glass 91] Personal communication, October, 1991.

[Good 83] Good, I. J., "The Philosophy of Exploratory Data Analysis," *Philosophy of Science*, Vol. 50, 283-295, 1983.

[Green 66] Green, D. and Swets, J. *Signal Detection Theory and Psychophysics*. Chapter 2. (New York, NY: John Wiley & Sons Inc., 1966.)

[Gupta 87] Gupta, V. N., Lennig, M. and Mermelstein, P., "Integration of Acoustic Information in a Large Vocabulary Word Recognizer," *Proc. ICASSP 87*, 697-700, April, 1987.

[Hetherington 91] Hetherington, I. L., Leung, H. C. and Zue, V. W., "Toward Vocabulary-Independent Recognition of Telephone Speech," *Proc. EuroSpeech 91*, 475-478, September, 1991.

[Hofstetter 92] Hofstetter, E., Rose, R, "Techniques for Task Independent Word Spotting in Continuous Speech Messages," *Proc. ICASSP 92*, II 101-104, March, 1992.

[Hon 89] Hon, H., Lee K., and Weidi, R., "Towards Speech Recognition Without Vocabulary-Specific Training," Presented at DARPA Speech and Natural Language Workshop, Philadelphia, PA, February, 1989.

[Huang 89] Huang, X. D. and Jack, M. A., "Semi-continuous Hidden Markov Models for Speech Recognition," *Computer, Speech and Language,*, Vol. 3, 1989.

[Hunt 89] Hunt, M. J. and Lefebvre, C., "A Comparison of Several Acoustic Representations for Speech Recognition with Degraded and Undegraded Speech," *Proc. ICASSP 89*, 262-265, May, 1989.

[Jelinek 76] Jelinek, F., "Continuous Speech Recognition by Statistical Methods," *IEEE Trans. on ASSP*, Vol. ASSP-64, 532-566, April, 1976.

[Jelinek 80] Jelinek, F. and Mercer, R. L., "Interpolated Estimation of Markov Source Parameters from Sparse Data," in *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal., (eds.), 381-397, (Amsterdam: North-Holland Publishing Co., 1980.)

[Johnson 88] Johnson, R. A. and Wichern, D. W., *Applied Multivariate Statistical Analysis, 2nd. ed.* (Englewood Cliffs, NJ: Prentice Hall, 1988.)

[Kassel 86] Kassel, R. H., "Aids for the Design, Acquisition, and Use of Large Speech Databases," S. B. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May, 1986.

[Katagiri 91] Katagiri, S, Lee, C. H., and Juang B. H., "New Discriminative Algorithms Based on the Generalized Probabilistic Descent Method," *Proc. IEEE-SP Workshop on Neural Networks for Speech Processing*, Princeton, Sept., 1991.

[Kawabata 89] Kawabata, T. and Shikano, K., "Island-Driven Continous Speech Recognizer Using Phone-Based HMM Word Spotting," *Proc. ICASSP 89*, 461-464, August, 1989.

[Kenny 90] Kenny, P., Lennig. M. and Mermelstein, P., "A Linear Predictive HMM for Vector-Valued Observations with Applications to Speech Recognition," *IEEE Trans. on ASSP*, Vol. ASSP-38, 220-225, February, 1990.

[Kewley-Port 82] Kewley-Port, D., "Measurement of Formant Transitions in Naturally Produced Stop Consonant-Vowel Syllables," *JASA*, Vol. 72, 379-389, 1982.

[Key 87] Key, K. M., "Acoustic Correlates of Place of Articulation in English Fricatives," S. M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1987.

[Klatt 76] Klatt, D. H., "Linguistic Uses of Segmental Duration in English: Acoustic and Perceptual Evidence," *JASA*, Vol. 59, 1208-1221, May, 1976.

[Klatt 82] Klatt,D. H., "Prediction of Perceived Phonetic Distance from Critical-Band Spectra: A First Step," *Proc. ICASSP 82*, 1278-1281, 1982.

[Klein 70] Klein, W., Plomp, R. and Pols, L. C. W., "Vowel Spectra, Vowel Spaces and Vowel Perception," *JASA*, Vol. 48, 999-1009, 1970.

[Kubala 90] Kubala, F., Schwartz, R. and Barry, C., "Speaker Adaptation from a Speaker-Independent Training Corpus," *Proc. ICASSP 90*, 137-140, April, 1990.

[Kubala 91] Kubala, F. and Schwartz, R., "A New Paradigm for Speaker-Independent Training," *Proc. ICASSP 91*, 833-836, May, 1991.

[Ladefoged 82] Ladefoged, P., *A Course in Phonetics, 2nd ed.*, (New York, NY: Harcourt Bruce Jovanovich), 1982.

[Lamel 86] Lamel, L. F., Kassel, R. H., & Seneff, S., "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus," *Proc. DARPA Speech Recognition Workshop,* Report No. SAIC-86/1546, 100-109, 1986.

[Lea 80] Lea, W. A. (ed.), *Trends in Speech Recognition,* (Englewood Cliffs, NJ: Prentice-Hall), 1980.

[Lee 88] Lee, K. F., "The SPHINX Speech Recognition System," Ph. D. Thesis, Computer Science Department, Carnegie Mellon University, April, 1988.

[Lee 89a] Lee, K. F., "Hidden Markov Models: Past, Present, and Future," *Proc. EuroSpeech 89*, 148-151, September, 1989.

[Lee 89b] Lee, K. F. and Hon, H. W., "Speaker-Independent Phone Recognition Using Hidden Markov Models," *IEEE Trans. on ASSP*, Vol. 37, 1641-1648, November, 1989.

[Lee 90] Lee, K. F., "Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition," *IEEE Trans. on ASSP*, Vol. 38, 599-609, April, 1990.

[LeMaire 89] LeMaire, V., Andre-Obrecht, R. and Jouvet, D., "An Acoustic-Phonetic Decoder Based on an Automatic Segmentation Algorithm," *Proc. EuroSpeech 89*, 392-395, September, 1989.

[Lisker 64] Lisker, L. and Abramson, A. S. "A cross- language study of voicing in initial stops: acoustical measurements," Word 20, 384-422, December, 1964.

[Leung 84] Leung, H. C. and Zue, V. W., "A Procedure for Automatic Alignment of Phonetic Transcriptions with Continuous Speech," *Proc. ICASSP 84*, 2.7.1-2.7.4, March, 1984.

[Leung 89] Leung, H. C., "The Use of Artificial Neural Networks for Phonetic Recognition," Ph. D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May, 1989.

[Leung 90] Leung, H. C., Glass, J. R., Phillips, M. S. and Zue, V. W., "Detection and Classification of Phonemes Using Context-Independent Error Back-Propagation," *Proceedings of International Conference on Spoken Language Processing,*: 1061-1064, Kobe, Japan, 1990.

[Leung 92] Leung, H. C., Hetherington, I. L. and Zue, V. W., "Speech Recognition Using Context-Dependent Stochastic Segment Neural Networks," *Proc. ICASSP 92*, I 613-616, March, 1992.

[Levinson 85] Levinson, S., "Structural Methods in Automatic Speech Recogniton," *Proceedings of the IEEE,* Vol. 73, No. 11, 1625-1650, November, 1985.

[Linde 80] Linde, Y., Buzo, A, and Gray R. M., "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communication*, Vol. COM-28, No. 1, 84-95, January, 1980.

[Lindgren 76] Lindgren, B. W., *Statistical Theory, 3rd. ed.* (New York, NY: MacMillan, 1976.)

[Lippmann 89] Lippmann, R., "Review of Neural Networks for Speech Recognition,", *Neural Computation* 1, 1989.

[Marcus 92] Marcus, J. N., "A Novel Algorithm for HMM Word Spotting, Performance Evaluation and Error Analysis," *Proc. ICASSP 92*, II 89-92, March, 1992.

[Meisel 91] Meisel, W. S., Anikst, M. T., Pirzadeh, J. E., Schumacher, J. E., Soares, M. C., and Trawick, D. J., "The SSI Large-Vocabulary Speaker-Independent Speech Recognition System,"*Proc. ICASSP 91*, 337-340, May, 1991.

[Meng 90] Meng, H. M. and Zue, V. W., "A Comparative Study of Acoustic Representations of Speech for Vowel Classification Using Multi-Layer Perceptrons," *Proceedings of the International Conference on Spoken Language Processing*, 1053-1056, November, 1990.

[Morrison 76] Morrison, D. F., *Multivariate Statistical Methods.* (New York: McGraw-Hill, 1976.)

[Mosteller 77] Mosteller, F. and Tukey, J. W., *Data Analysis and Regression.* (Reading, MA: Addison-Wesley, 1977.)

[Myers 88] Myers, R. H., *Classical and Modern Regression with Applications.* (Boston, MA: Duxbury Press, 1988.)

[Nadas 84] Nádas, A., "Estimation of Probabilities in the Language Model of the IBM Speech Recognition System," *IEEE Trans. on ASSP*, Vol. ASSP-32, 859-861, August, 1984.

[Ney 90] Ney, H., "Experiments on Mixture-Density Phoneme-Modelling for the Speaker-Independent 1000-Word Speech Recognition DARPA Task," *Proc. ICASSP 90*, 713-716, April, 1990.

[Niyogi 91] Niyogi, P. and Zue, V. W., "Correlation Analysis of Vowels and their Application to Speech Recognition," *Proc. EuroSpeech 91*, 1253-1256, September, 1991.

[Ostendorf 89] Ostendorf, M. and Roukos, S., "A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition," *IEEE Trans. on ASSP*, Vol. ASSP-37, 1857-1869, December, 1989.

[Pallett 91] Pallett, D., "DARPA Resource Management and ATIS Benchmark Test Poster Session," Presented at DARPA Workshop on Speech and Natural Language, Pacific Grove, CA, February, 1991.

[Paul 88] Paul, D. B., Martin, E. A., "Speaker Stress-Resistant Continuous Speech Recognition," *Proc. ICASSP 88*, 283-286, September, 1988.

[Paul 91] Paul, D., "The Lincoln Tied-Mixture HMM Continuous Speech Recognizer," *Proc. ICASSP 91*, 329-332, May, 1991.

[Peeling 91] Peeling, S. M. and Ponting K. M., "Variable Frame Rate Analysis in the ARM Continuous Speech Recognition System," *Speech Communication*, **10**, 155-162, 1991.

[Peterson 52] Peterson, G. E. and Barney, H. L., "Control Methods Used in a Study of Vowels," *JASA*, Vol. 24, 175-184, 1952.

[Phillips 92] Phillips, M., Glass, J., Polifroni J. and Zue, V. W., "Collection and Analyses of WSJ-CSR Data at MIT," Presented at DARPA Workshop on Speech and Natural Language, Harriman, NY, February, 1992.

[Plomp 67] Plomp, R., Pols, L. C. W. and van de Geer, J. P., "Dimensional Analysis of Vowel Spectra," *JASA*, Vol. 41, No. 3, 707-712, 1967.

[Pols 69] Pols, L. C. W., Kamp, L. J. T. and Plomp, R., "Perceptual and Physical Space of Vowel Sounds," *JASA*, Vol. 53, No. 4, 1093-1101, 1973.

[Pols 73] Pols, L. C. W., Tromp, H. R. C. and Plomp, R., "Frequency Analysis of Dutch Vowels from 50 Male Speakers," *JASA*, Vol. 53, No. 4, 1093-1101, 1973.

[Rabiner 83] Rabiner, L. R., Levinson, S. E., and Sondhi, M. M., "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition", *BSTJ*, Vol. 62, 1075-1106, April, 1983.

[Rabiner 89] Rabiner, L. R., Wilpon J. G. and Soong, F. K., "High Performance Connected Digit Recognition Using Hidden Markov Models," *IEEE Trans. on ASSP*, Vol. ASSP-37, 1214-1223, August, 1989.

[Renals 92] Renals, S., Morgan, N., Cohen, M. and Franco, H., "Connectionist Probability Estimation in te DECIPHER Speech Recognition System," *Proc. ICASSP 92*, pp. I 601-604, March, 1992.

[Rigoll 89] Rigoll, G., "Speaker Adaptation for Large Vocabulary Speech Recognition System using 'Speaker Markov Models'," *Proc. ICASSP 89*, 5-8, May, 1989.

[Robinson 91a] Robinson, T. and Fallside, F., "A Recurrent Error Propagation Network Speech Recognition System," *Computer, Speech and Language,*, Vol. 5, No. 3, July, 1991.

[Robinson 91b] Robinson, T., "Several Improvements to a Recurrent Error Propagation Network Phone Recognition System," Cambridge Universtity Technical Report CUED/F-INFENG/TR82, September, 1991.

[Rohlicek 89] Rohlicek, J. R., Russell, W., Roukos, S., and Gish, H., "Continuous Hidden Markov Modeling for Speaker-Independent Word Spotting," *Proc. ICASSP 89*, 627-630, May, 1989.

[Rohlicek 92] Rohlicek, J.R., Ayuso, D., Bates, M., Bobrow, R., Boulanger, A., Gish, J., Jeanrenaud, P., Meteer, M., and Siu, M., "Gisting Conversational Speech," *Proc. ICASSP 92*, II 113-116, March, 1992.

[Rose 90] Rose, R.C. and Paul. D.B., "A Hidden Markov Model Based Keyword Recognition System," *Proc. ICASSP 90*, 129-132, May, 1990.

[Rose 91] Rose, R. C., Chang, E. I., and Lippmann, R. P., "Techniques for Information Retrieval from Voice Messages," *Proc. ICASSP 91*, 317-320, April, 1991.

[Rtischev 89] Rtischev, D., "Speaker Adaptation in a Large Vocabulary Speech Recognition System," S. M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, January, 1989.

[Rudnicky 87] Rudnicky, A. I., Baumeister, L. K., DeGraaf, K. H., and Lehmann, E., "The Lexical Access Component of the CMU Continuous Speech Recognition System," *Proc. ICASSP 87*, 376-379, April, 1987.

[Schafer 77] Schafer, R. and Markel, J. (eds.), *Speech Analysis*, (New York: IEEE Press, 1977.)

[Schwartz 84] Schwartz, R., Chow, Y., Roucos, S., Krasner, M., and Makhoul, J., "Improved Hidden Markov Modeling of Phonemes for Continuous Speech Recognition," *Proc. ICASSP 84*, Paper 35.6, May, 1984.

[Schwartz 85] Schwartz, R., Chow, Y., Kimball, O., Roucos, S., Krasner, M., and Makhoul, J., "Context- Dependent Modelling for Acoustic-Phonetic Recognition of Continuous Speech," *Proc. ICASSP 85*, 1205-1208, August, 1985.

[Schwartz 91] Schwartz, R. and Austin, S., "A Comparison of Several Approximate Algorithms for Finding Multiple (N-BEST) Sentence Hypotheses," *Proc. ICASSP 91*, 701-704, May, 1991.

[Schwartz 92] Schwartz, R., Austin, S., Kubala, F., Makhoul, J., Nguyen, L., Placeway, P., Zavaliagkos, G., "New Uses for the N-Best Sentence Hypotheses within the BYBLOS Speech Recognition System," *Proc. ICASSP 92*, I 1-4, March, 1992.

[Seneff 86] Seneff, S. "An Auditory-Based Speech Recognition Strategy: Application to Speaker-Independent Vowel Recognition," Proceedings of Speech Recognition Workshop, Palo Alto, CA, February 19-20, 1986.

[Seneff 88] Seneff, S., "A Joint Synchrony/Mean Rate Model of Auditory Speech Processing," *Journal of Phonetics*, Vol. 16, no. 1, 55-76, 1988.

[Soclof 90] Soclof, M. and Zue, V. W., "Collection and Analysis of Spontaneous and Read Corpora for Spoken Language System Development," *Proceedings of International Conference on Spoken Language Processing,*: 1105-1108, Kobe, Japan, 1990.

[Soli 81] Soli, S. D., "Second Formants in Fricatives: Acoustic Consequences of Fricative-Vowel Coarticulation," *JASA*, Vol. 70, 976-984, October, 1981.

[Sorensen 89] Sorensen, H. B. D. and Dalsgaard, P., "Multi-Level Segmentation of Natural Continuous Speech Using Different Auditory Front-Ends," *Proc. EuroSpeech 89*, 79-82, September, 1989.

[Stern 87] Stern, R. M. and Lasry, M. J., "Dynamic Speaker Adaptation for Feature-Based Isolated Word Recognition," *IEEE Trans. on ASSP*, Vol. ASSP-37, 751-763, June, 1987.

[Stern 91] Rozzi, W. A. and Stern, R. M., "Speaker Adaptation in Continuous Speech Recognition via Estimation of Correlated Mean Vectors," *Proc. ICASSP 91*, 865-868, May, 1991.

[Stevens 78] Stevens, K. N. and Blumstein, S. E., "Invariant Cues for Place of Articulation of Stop Consonants," *JASA*, Vol. 64, 1358-1368, 1978.

[Stevens 85] Stevens, K. N., "Evidence for the role of acoustic boundaries in the perception of speech sounds," in *Phonetic Linguistics: Essays in Honor of Peter Ladefoged*, V. Fromkin, (ed.), 243-255, (New York: Academic Press, 1985.)

[Stevens 86] Stevens, K. N., "Models of Phonetic Recognition II: An Approach to Feature-Based Recognition," *Proceedings of the Montreal Symposium of Speech Recognition*, 67-68, July, 1986.

[Stevens 87] Stevens, K. N., Course notes for 6.541J, M.I.T., Spring, 1987.

[Stevens 90] Stevens, K. N., Personal communication.

[Sussman 91] Sussman, H. M., McCaffrey, H. A., and Matthews, S. A., "An Investigation of Locus Equations as a source of Relational Invariance for Stop Place Categorization," *JASA*, Vol. 90, 1309-1325, September, 1991.

[Tukey 86] Tukey, J. W. and Wilk, M. B., "Data Analysis and Statistics: An Expository Overview," in *The Collected Works of John W. Tukey, Volume IV, Philosophy and Principles of Data Analysis: 1965-1986*, L. V. Jones, (ed.), 549-578, (Monterrey, CA: Wadsworth & Brooks/Cole, 1986.)

[Tukey 77] Tukey, J.W, *Exploratory Data Analysis.* (Reading, MA: Addison-Wesley, 1977.)

[Vicenzi 86] Vicenzi, C. and Sciarra, D., "Using Diphones in Large Vocabulary Word Recognition," *Proceedings of the Montreal Symposium of Speech Recognition*, 47-50, July, 1986.

[Viterbi 67] Viterbi, A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. on Information Theory*, **IT-13**, 260-269, 1967.

[Waibel 88] Waibel, A., Hanazawa, T., Shikano, K., and Lang, K., "Phoneme recognition: neural networks vs. hidden Markov models," *Proc. ICASSP-88,* 107-110, April, 1988.

[Weintraub 87] Weintraub, M. and Bernstein, J., "RULE: A System for Constructing Recognition Lexicons," Presented at DARPA Speech Recognition Workshop, San Diego, CA, March, 1987.

[Weintraub 89] Weintraub, M., Murveit, H., Cohen, M., Price P., Bernstein J., Baldwin, G., and Bell, D., "Linguistic Constraints in Hidden Markov Model Based Speech Recognition," *Proc. ICASSP 89*, 699-702, May, 1989.

[Wilcox 92] Wilcox, L. and Bush,M., "Training and Search Algorithms for an Interactive Wordspotting System," *Proc. ICASSP 92*, II 97-101, March, 1992.

[Wilpon 90] Wilpon, J.G., Rabiner, L.R., Lee, C.-H., and Goldman, E.R., "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models," *IEEE Trans. on ASSP*, **Vol. ASSP-38, No. 11**, 1870-1878, November, 1990.

[Wilpon 91] Wilpon, J.G., Miller, L.G., and Modi, P., "Improvements and Applications for Key Word Recognition Using Hidden Markov Techniques," *Proc. ICASSP 91*, 309-312, May, 1991.

[Zue 85] Zue, V. W., "The Use of Speech Knowledge in Automatic Speech Recogniton," *Proceedings of the IEEE,* Vol. 73, No. 11, 1602-1614, November, 1985.

[Zue 89a] Zue, V., Glass, J., Phillips, M., and Seneff, S., "The MIT SUMMIT Speech Recognition System: A Progress Report" Presented at DARPA Speech and Natural Language Workshop, Philadelphia, PA, February, 1989.

[Zue 89b] Zue, V., Glass, J., Phillips, M., and Seneff, S., "Acoustic Segmentation and Phonetic Recognition in the SUMMIT System," *Proc. ICASSP 89*, 389-392, May, 1989.

[Zue 89c] Zue, V., Seneff, S. and Glass, J., "Speech Database Development: TIMIT and Beyond," Presented at Workshop on Speech Input/Output Assessment and Speech Databases, Amsterdam, the Netherlands, September, 1989.

[Zue 90a] Zue, V., Glass, J., Goodine, D.., Phillips, M., and Seneff, S., "The SUMMIT Speech Recognition System: Phonological Modelling and Lexical Access," *Proc. ICASSP 90*, 49-52, April, 1990.

[Zue 90b] Zue, V., Glass, J., Phillips, M., and Seneff, S., "Recent progress on the SUMMIT System," Presented at DARPA Speech and Natural Language Workshop, Hidden Valley, PA, June, 1990.

[Zue 92] Personal communication.

[Zwicker 61] Zwicker, E., "Subdivision of the Audible Frequency Range into Critical Bands (frequenzgruppen)," *JASA*, Vol. 33, 248-249, 1961.

Appendix A

Determining Weights in Maximum Spectral Deviation Computation

This appendix outlines the method used for determining the weights in the maximum spectral deviation coefficient computation described in Section 3.4.4. As described in that section, the expression

$$\hat{\tau} = \arg\max_{\tau} \sum_{i=1}^{q} w_i (h_{i\tau} - \hat{h}_{i\tau})^2$$

is used to determine the frame $\hat{\tau}$ to be used for computing the maximum spectral deviation coefficients, where q is the number of hair cell envelope principal components (HCEPC's) used in the computation, $h_{i\tau}$ is the i^{th} HCEPC measured at frame τ, $\hat{h}_{i\tau}$ is the linearly interpolated estimate of $h_{i\tau}$, determined as in Equation 3.7, and w_i, $1 \leq i \leq q$ are the weights, which are used to account for the fact that the scale of $(h_{i\tau} - \hat{h}_{i\tau})^2$ for $\tau = \hat{\tau}$ is dependent on i, the index of the HCEPC.

For each HCEPC i, the weight w_i used was $1/{s_i}^2$ where s_i is an estimate of the standard deviation of $(h_{i\hat{\tau}} - \hat{h}_{i\hat{\tau}})$. Making this estimate by computing the above expression over a sample of segments and using the sample standard deviation is complicated by the fact that the estimate requires $\hat{\tau}$ to be known. However, $\hat{\tau}$ depends on the weights themselves so making a direct estimate is impossible. Instead, for each segment in the sample used to make the estimate, the maximum spectral deviation for each component i was used in making the estimate for that component. Formally, for each segment j in the sample, the value

$$v_{ij} = \max_{\tau,\ 1 \leq \tau \leq T_j} |h_{ij\tau} - \hat{h}_{ij\tau}|$$

was used in determining s_i, where T_j is the number of frames in segment j, $h_{ij\tau}$ is the value of the i^{th} HCEPC for frame τ on segment j and $\hat{h}_{ij\tau}$ is the value of the linearly interpolated estimate of $h_{ij\tau}$. While the sample standard deviation of v_{ij} over all j could be used in making the estimate, doing so would ignore the statistical dependence of v_{ij} on T_j, the length of the segment. In particular, it stands to reason that the typical maximum spectral deviation for any component should increase with segment length since for a longer segment, there should tend to be a larger spectral change over the course of the segment. If this effect is ignored, the sample standard deviations might be dominated by the large values of v_{ij} obtained for the longer segments. Thus, the determination of the set of s_i was more involved. In particular, the algorithm for computing the set is:

1. For each i and each segment length t, $4 \leq t < 100$, where t is measured in number of frames, compute

   $$\rho_{it} = [\frac{1}{N(T_j = t)} \sum_{\text{all } j,\ T_j = t} {v_{ij}}^2 - (\frac{1}{N(T_j = t)} \sum_{\text{all } j,\ T_j = t} v_{ij})^2]^{\frac{1}{2}}$$

   where $N(T_j = t)$ is the number of segments in the sample consisting of t frames. Thus, ρ_{it} is an estimate of the standard deviation in v_{ij} as a function of t. For $t = 100$, ρ_{it} is computed in the same manner but all segments j for which $T_j \geq 100$ are used in the computation. There were no segments of fewer than four frames so there were no estimates made for $t < 4$.

2. For $2 \leq i \leq q$, set s_i to be the slope of the least squares fit through the

origin of the set of ρ_{it} to the set of ρ_{1t} multiplied by s_1. Thus,

$$s_i = s_1 \frac{\sum_{t=1}^{100} \rho_{1t}\rho_{it}}{\sum_{t=1}^{100} {\rho_{it}}^2}.$$

The slope computed for component i can be interpreted as an estimate of the ratio of the standard deviation of v_{ij} to that of the standard deviation of v_{1j} averaged over all segment lengths.

3. Finally, as stated above, set $w_i = 1/{s_i}^2$, $1 \leq i \leq q$.

The segmenter that was eventually used in our work includ d the maximum spectral deviations MSD$_i$ for $1 \leq i \leq 3$. The proportions oi the empirically determined weights for these components were w_1 = 1.0, $w_2 = .66$ and $w_3 =$ .51, confirming our hypothesis that the weights would decrease with i.

Appendix B

Estimating Bigram Transition Probabilities

This appendix describes the method we use to estimate initial subword model state probabilities and transition probabilities between subword models for use in the phonetic recognition task described in Chapter 4 and the word spotting task of Chapter 5. For the purposes of this discussion, we introduce the term *segment sequence label* (SSL) to refer to the phone or biphone label associated with a sequence of segments by the segmenter. For example, if the segmenter produces a segment sequence associated with the biphone $/B//C/$, where $/B/$ and $/C/$ are variables that represent phonetic transcription labels, we will refer to this event as an occurrence of the SSL B–C. As described in Chapter 3, any model x is trained using segment sequences labelled x. We will continue to follow the convention just introduced, distinguishing variables used to represent SSL's and model labels from those used to represent phonetic transcription labels by enclosing the latter in slashes (/) and using uppercase letters to refer to components of SSL and model names and lowercase letters to refer to complete SSL and model labels, which may represent phones or biphones.

The usual bigram estimate for the transition probability $\hat{\Pr}(x \rightarrow y)$ from model x to model y can be expressed as

$$\hat{\Pr}(x \rightarrow y) = \frac{N(x \rightarrow y)}{N(x)} \tag{B.1}$$

where $N(x \rightarrow y)$ is the number of times that the SSL sequence xy occurs in the training set and $N(x)$ is the count of SSL x.

In a frame-based HMM system, estimating the bigram transition probabilities is straightforward since each model represents a single phone. Thus, Eq. B.1 can be computed simply using the training set phonetic transcriptions for computing phone sequence counts. For instance,

$$\hat{\Pr}(A \rightarrow B) = \frac{N(/A/ \rightarrow /B/)}{N(/A/)}$$

where $N(/A/ \rightarrow /B/)$ is the number of times phone $/B/$ follows $/A/$ and $N(/A/)$ is the count of $/A/$ in the training set. It is easy to generalize this method for estimating the $\pi(B)$, the initial state probability for model B, by using the artifice of starting each utterance with the label /BEGIN/. Then

$$\pi(B) = \frac{N(\text{BEGIN} \rightarrow B)}{N(\text{BEGIN})}.$$

We could directly apply Eq. B.1 to estimate transition probabilities in the segment-based HMM. The problem with this simple approach is that there may not be sufficient training data for estimating some of these transition probabilities. In particular if x or y is a biphone SSL, both the equation's numerator and denominator may be too small to obtain a good estimate.

For instance, for SSL A to be followed by SSL B–C,

1. phone $/A/$ must be followed by phone $/B/$,

2. phone $/B/$ must be followed by phone $/C/$, and

3. the segmenter must produce SSL B–C from the phone sequence $/B//C/$.

This event may be too rare to allow for a good estimate of $\hat{\Pr}(A \rightarrow B\text{–}C)$.

We deal with this problem by assuming that the process for generating phone sequences is independent of segmenter behavior. By decomposing the problem in this way, we estimate the probability of the above event as the product of probability estimates for each of the three subevents just described, none of which is as rare as the occurrence of SSL B–C after SSL A. A bigram model is used to estimate each event.

To be precise, say we are estimating $\hat{\Pr}(D\text{–}A \rightarrow B\text{–}C)$, where D and C may each be a null label, i.e, if D is a null label, $D\text{–}A = A$. For clarity we will assume that all SSL's are of length one or two, although this assumption is not important in the following description. Our segmenter rarely produces SSL's of length greater than two in any case.

Define a *language* training set $\mathcal{L}$ and a segmenter training set S. The former is used to estimate phone sequence probabilities and the latter is used to model segmenter behavior. The language training set consists of the ten male VOYAGER training speakers used to train the acoustic models as well as sixteen female VOYAGER speakers. These data are used since the phone sequence counts in these utterances should be similar to those in the test set, which are also VOYAGER utterances. The segmenter training set consists of the male TIMIT and VOYAGER speakers used to train the acoustic models. By using a distinct training set for each purpose, we have access to more training data than if both training sets consisted of the male VOYAGER speakers.

Let $\hat{N}_{\mathcal{L}}(D\text{–}A \rightarrow B\text{–}C)$ be an estimate of the number of times that SSL D–A is followed by B–C in $\mathcal{L}$. Note that we cannot compute the actual number because $\mathcal{L}$ consists of female VOYAGER speakers whose utterances have not been run through the segmenter. Let $\hat{N}_{\mathcal{L}}(D\text{–}A)$ be an estimate of the count of SSL D–A in $\mathcal{L}$. Then the formula

$$\hat{\Pr}(D\text{–}A \rightarrow B\text{–}C) = \frac{\hat{N}_{\mathcal{L}}(D\text{–}A \rightarrow B\text{–}C)}{\hat{N}_{\mathcal{L}}(D\text{–}A)} \qquad \text{(B.2)}$$

is used to estimate the transition probability. First, we assume that the probability is independent of D. This is in keeping with the spirit of bigram estimates since D is two labels to the left of B and bigram estimates only depend on the identities of adjacent labels. Thus, the above equation is valid for any SSL ending with label A.

Using this assumption, we estimate the denominator of Eq. B.2 as follows. We note that an SSL u–A ending in A is produced for each instance of phone $/A/$ except when A is merged on the right into some SSL A–v, v being a non-null label. Thus, letting $\hat{\Pr}(u\text{–}A|\ /A/)$ be the estimated probability that A is *not* merged on the right and $N_{\mathcal{L}}(/A/)$ be the count of $/A/$ in the language training set,

$$\hat{N}_{\mathcal{L}}(D\text{–}A) = N_{\mathcal{L}}(/A/)\hat{\Pr}(u\text{–}A|\ /A/). \qquad \text{(B.3)}$$

The segmenter training set $\mathcal{S}$ is used to compute $\hat{\Pr}(u\text{–}A|/A/)$. Letting $N_{\mathcal{S}}(u\text{–}A)$ be the number of instances of SSL's ending in A and $N_{\mathcal{S}}(/A/)$ be the count of phone $/A/$ in the segmenter training set,

$$\hat{\Pr}(u\text{–}A|\ /A/) = \frac{N_{\mathcal{S}}(u\text{–}A)}{N_{\mathcal{S}}(/A/)}. \qquad \text{(B.4)}$$

Thus, the denominator of Eq. B.2 can be computed by combining Equations B.3 and B.4.

Similar reasoning is used to compute the numerator. An SSL that starts with B follows one ending in A each time $/B/$ follows $/A/$ and the two are not merged into a single SSL A–B. Similar to above, the formula for computing $\hat{N}_{\mathcal{L}}(u\text{–}A \rightarrow B\text{–}v)$, an estimate of the number of times such an SSL sequence occurs in training set $\mathcal{L}$ is

$$\hat{N}_{\mathcal{L}}(u\text{–}A \rightarrow B\text{–}v) = N_{\mathcal{L}}(/A/ \rightarrow /B/)(1 - \frac{N_{\mathcal{S}}(A\text{–}B)}{N_{\mathcal{S}}(/A/ \rightarrow /B/)}) \qquad \text{(B.5)}$$

where $N_{\mathcal{L}}(/A/ \rightarrow /B/)$ is the number of times $/B/$ follows $/A/$ in the language training set, $N_{\mathcal{S}}(A\text{–}B)$ is the count of SSL A–B in the segmenter training set and $N_{\mathcal{S}}(/A/ \rightarrow /B/)$ is the number of times $/B/$ follows $/A/$ in the segmenter training set.

Given this estimate of the number of times an SSL starting with B follows SSL D–A in the language training set, the estimated number of instances that SSL D–A is followed by SSL B–C is given by

$$\hat{N}_{\mathcal{L}}(D\text{–}A \rightarrow B\text{–}C) = \hat{N}_{\mathcal{L}}(u\text{–}A \rightarrow B\text{–}v)\hat{\Pr}(B\text{–}C|B\text{–}v)$$

where $\hat{\Pr}(B\text{–}C|B\text{–}v)$ is the estimated probability that given that an SSL starts with B, the it is $B\text{–}C$. The formula for this is

$$\hat{\Pr}(B\text{–}C|B\text{–}v) = \frac{N_S(B\text{–}C)}{N_S(B\text{–}v)} \tag{B.6}$$

where the numerator in Equation B.6 is the count of SSL $B\text{–}C$ and the denominator is the the count of all SSL's starting with B in the segmenter training set.

Combining Eqs. B.2-B.6, we have

$$\hat{\Pr}(D\text{–}A \rightarrow B\text{–}C) = \frac{N_{\mathcal{C}}(/A/ \rightarrow /B/)(1 - \frac{N_S(A\text{–}B)}{N_S(/A/ \rightarrow /B/)})N_S(B\text{–}C)N_S(/A/)}{N_{\mathcal{C}}(/A/)N_S(B\text{–}v)N_S(u\text{–}A)} \tag{B.7}$$

and this is used to estimate transition probabilities between models. Initial model state probabilities are estimated using the same formula using the artifice of beginning each utterance with the label /BEGIN/ as described above.

Appendix C

Guide to Abbreviations

On the following page is a table of abbreviations of terms that are used throughout the thesis. Note that abbreviations for most of the segmental measurements referred to in Chapter 4 are defined in Tables 4.2 and 4.3.

Abbreviation	Meaning
CG	center of gravity
EDA	exploratory data analysis
GMDA	grouped multiple discriminant analysis
GTI	garbage trial interval
HCE	hair-cell envelopes
HCEPC	hair-cell envelope principal components
HMM	hidden Markov model
IPA	International Phonetic Alphabet
KTI	keyword trial interval
LOR	log odds ratio
LPC	linear predictive coding
MDA	multiple discriminant analysis
MFSC	mel-frequency spectral coefficient
MFSPC	mel-frequency spectral principal components
MLS	multi-level segmentation
MSD	maximum spectral deviation
MSDS	maximum spectral deviation spectrum
MSE	mean squared error
PDF	probability distribution function
RMS	root mean square
RMSE	root mean squared error
ROC	receiver operating curve
SSL	segment sequence label
VOT	voice onset time
VQ	vector quantization

Table C.1: Table of abbreviations.

DARPA OFFICIAL DISTRIBUTION LIST

DIRECTOR 2 copies
Information Processing Techniques Office
Defense Advanced Research Projects Agency (DARPA)
1400 Wilson Boulevard
Arlington, VA 22209

OFFICE OF NAVAL RESEARCH 2 copies
800 North Quincy Street
Arlington, VA 22217
Attn: Dr. Gary Koop, Code 433

DIRECTOR, CODE 2627 6 copies
Naval Research Laboratory
Washington, DC 20375

DEFENSE TECHNICAL INFORMATION CENTER 2 copies
Cameron Station
Alexandria, VA 22314

NATIONAL SCIENCE FOUNDATION 2 copies
Office of Computing Activities
1800 G. Street, N.W.
Washington, DC 20550
Attn: Program Director

HEAD, CODE 38 1 copy
Research Department
Naval Weapons Center
China Lake, CA 93555